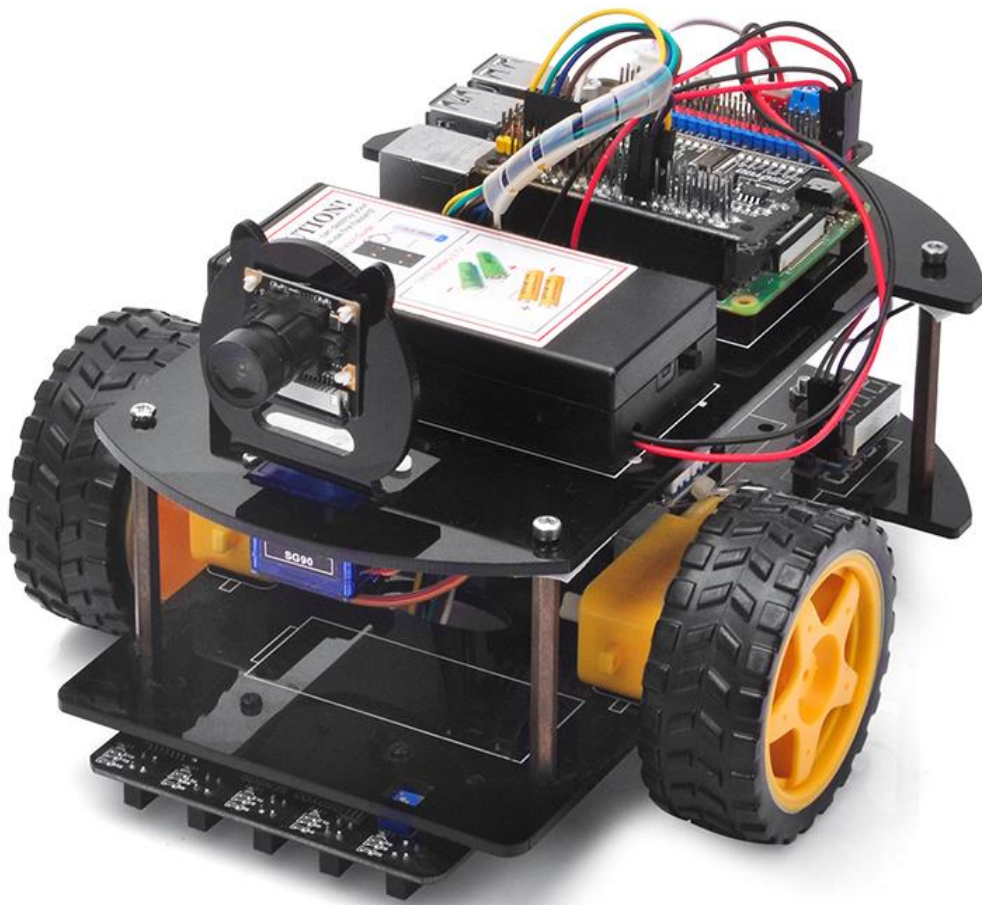


## OSOYOO Robotic Car for Raspberry Pi

# User manual



Online Tutorial Index: <https://osoyoo.com/?p=52861>

If you have any problem, please feel free to contact us, our email address is [support@osoyoo.info](mailto:support@osoyoo.info)

## Content

OSOYOO Robotic Car for Raspberry Pi .....	1
User manual .....	1
<b>Preface</b> .....	1
Safety Instruction .....	1
About OSOYOO .....	1
About This Kit .....	1
Online Tutorial Index: .....	2
Customer Service and Tech Support .....	2
<b>Lesson 0: Introduction</b> .....	3
Parts and Devices .....	5
Tutorial and sample projects .....	7
<b>Lesson 1: Basic Installation and Movement</b> .....	8
Objective .....	8
Parts and Devices .....	8
Hardware Installation .....	11
Circuit Connection .....	15
Software Installation .....	19
<b>Lesson 2: Line Tracking (Python)</b> .....	35
Objective .....	35
Parts and Devices .....	35
Hardware Installation .....	35
Circuit Connection .....	36
Software Installation .....	38
<b>Lesson 3: Obstacle Avoidance</b> .....	40
Objective .....	40
Parts and Devices .....	40
Hardware Installation .....	41
Circuit Connection .....	42
Software Installation .....	43
<b>Lesson 4: Use Mobile APP to control with UDP (Python)</b> .....	46
Objective .....	46
Parts and Devices .....	46
Hardware Installation .....	47

Software Installation .....	48
Lesson 5: Make a simple website server in Pi (Python).....	54
Objective.....	54
Parts and Devices.....	54
Hardware Installation .....	55
Software Installation .....	55
Lesson 6: Web-Camera-Controlled IoT Raspberry Pi Robot Car (Python) ....	60
Objective.....	60
Parts and Devices.....	60
Hardware Installation .....	61
Circuit Connection .....	63
Software Installation .....	64
Start jpeg-streamer server in your Raspberry Pi.....	65
Web-Camera-Controlled IoT Robot Car .....	68

# Preface

## Safety Instruction

- Please turn off battery power when your Robot is connected to Personal Computer or Laptop via USB cable
- Do not plug or unplug and wire or module when power is on,
- Do not even touch the board when power is on or your hand has static charge (you can move static charge by touching metal tap).
- Reversing +/- when connecting modules with OSOYOO board for Arduino, or incorrectly connecting can destroy your electronics and cause fire hazard.
- Please make sure polar direction is correct when you install batteries in your battery box, otherwise it can destroy your device and cause fire hazard.
- Please don't use Carbon zinc batteries as power, as output current of this kind of battery is too low and it is unable to load the robot car
- Do not leave batteries in battery box if you don't use it for long time due to the risk of fire and malfunction.

## About OSOYOO

OSOYOO brand owned by Pinetree Electronics Ltd, the only Canadian Owned Science Fair Supplier, Circuit Builder, IOT, Autonomous Smart Machine, Home Electronics Workshop and Arduino Accessory Company.

Pinetree Electronics Ltd established since 2009 in Vancouver, Canada. We are Engineers and Programmers Ourselves, So We Know How Important It Is to CREATE Your Dreams! Not All Robot Kits Are Created Equal: Pay For True Quality & You Will Not Be Disappointed. For more information and tutorial of OSOYOO products, please visit [www.osoyoo.com](http://www.osoyoo.com)

## About This Kit

OSOYOO robot car for Arduino is a perfect combination of challenge and excitement,



learning and fun. The kit comes with step by step ONLINE tutorial with text, picture and video. The kit is great for any skill level — whether you're a pro, enthusiast, or a beginner.

## Online Tutorial Index:

- General Introduction: <https://osoyoo.com/?p=52861>
- OSOYOO robot car introduction video: <https://osoyoo.com/manual/2020005500.html>
- OSOYOO robot car tutorial PDF download link:  
<https://osoyoo.com/manual/2020005500.pdf>

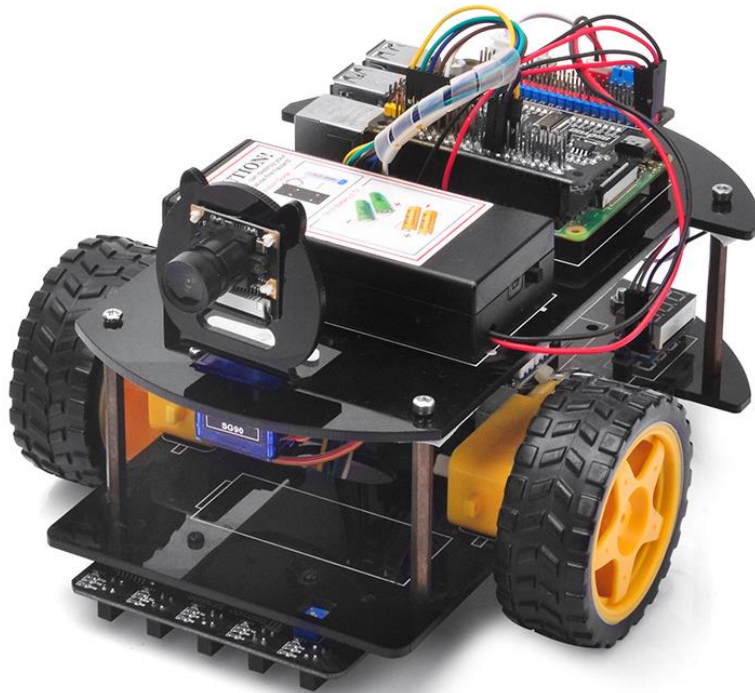
## Customer Service and Tech Support

You have following two options to ask for help:

- 1) Leave comments on our tutorial <https://osoyoo.com/?p=52861>
- 2) Send email to [support@osoyoo.info](mailto:support@osoyoo.info). We guarantee that all inquiries will be replied in 24 hours.

# OSOYOO Robot car for raspberry pi

## Lesson 0: Introduction



There are many entry level Robot Car Kits in the market, most of them are controlled by Arduino Boards. You can check our tutorial blog for such OSOYOO Robot kit for Arduino in: <https://osoyoo.com/?p=32220>

The advantage of Arduino Robot Car kit is that Arduino has no Operating System and programming is simple and easily. For some basic robot application which needs only simply logic to handle sensor data and control actuators, Arduino-controlled robot car is a good choice.

However, for some more complex robot applications which need more complex functions such as computer vision(CV), Internet of Things (IoT), web server control etc, Arduino board's ability is too weak to reach the target.



In order to help intermediate students to complete some complex Robotic project. We developed a more advanced Raspberry Pi Robot Car learning Kit.

Why Raspberry Pi is so important to the Robot Car DIY learning kit?

Because Raspberry Pi is a real computer which has Linux OS (Raspberry Pi OS) and therefore much powerful than Arduino Board which is simply a microcontroller(MCU).

With Raspberry Pi OS and its huge open-source software community, people can make much complicated Robot projects, i.e. web application, database, A.I., machine learning, IoT, Computer Vision etc.

Unlike Arduino board, Raspberry Pi programming environment is much more complex and flexible. It supports almost all programming language as long as the language is supported by Raspberry Pi OS Open Source community. The most commonly used languages to access Raspberry Pi GPIO pins are C and Python. If you want to learn some Raspberry Pi hardware GPIO programming, you can read our tutorial in following links:

<https://osoyoo.com/?p=11587>

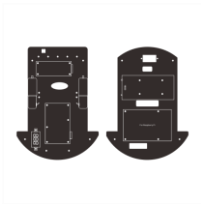



Since Raspberry Pi programming is different from Arduino, we suggest user should get some basic Linux knowledge before practice Raspberry Pi Robot project. If you want to learn Linux Robot by doing, you can buy this kit as learning kit. We provide some basic lessons in the kit to show you how to set up a Raspberry Pi OS Linux environment, how to use a Linux shell to control car movement and how to use a web browser to monitor and control car movement through the camera.

These basic projects in our tutorial have step-by-step instruction with sample code, circuit graph and installation video. These lessons all have been tested, so if you strictly follow our instruction, they will work without any problem. We also write detailed comments in our python sample code, which can help you to understand the code and customize for your own application. However, you must have some Linux and python background knowledge, otherwise please do not change the code.

We also provide an optional Open-CV sample project for use to practice Robotic Computer Vision procedure. However, as OpenCV and machine learning open source community is evolving every day, we can not guarantee that the OpenCV project will work properly. You might do your due diligence to follow up the OpenCV community's updates and make your project work.

If you have any interesting application to use our Robot kit, you are more than welcome to share your excellent ideas in our comment section.

## Parts and Devices

No.	Picture	Product Name	Link
1		Pie car chassis	<a href="#">click to buy</a>
2		Gear motor	<a href="#">click to buy</a>
3		Wheel	<a href="#">click to buy</a>
4		Universal Wheel	<a href="#">click to buy</a>



5		<a href="#">OSOYOO PWM HAT</a>	<a href="#">click to buy</a>
6		<a href="#">OSOYOO model X motor driver module</a>	<a href="#">click to buy</a>
7		CSI camera with cable	<a href="#">click to buy</a>
8		camera holder	<a href="#">click to buy</a>
9		Tracking sensor module	<a href="#">click to buy</a>
10		Ultrasonic sensor	<a href="#">click to buy</a>
11		Ultrasonic sensor holder	<a href="#">click to buy</a>

12		Servo motor	<a href="#">click to buy</a>
13		Voltage meter	<a href="#">click to buy</a>
14		18650 battery case	<a href="#">click to buy</a>
15		18650 battery charger	<a href="#">click to buy</a>
16		2000mAh 18650 rechargeable battery	<a href="#">click to buy</a>

## Tutorial and sample projects

- Lesson 1: <https://osoyoo.com/?p=51393>
- Lesson 2: <https://osoyoo.com/?p=52867>
- Lesson 3: <https://osoyoo.com/?p=52874>
- Lesson 4: <https://osoyoo.com/?p=52879>
- Lesson 5: <https://osoyoo.com/?p=52886>
- Lesson 6: <https://osoyoo.com/?p=52891>

# OSOYOO Robot Car for Raspberry Pi

## Lesson 1: Basic Installation and Movement

### OBJECTIVE

Welcome to the first lesson of OSOYOO Robot Car for Raspberry Pi!



In this lesson, we will introduce how to install the framework of the OSOYOO Robot Car for Raspberry Pi firstly, then we will simply tell you how to install Raspberry Pi OS for the Raspberry pi and how to use console to control raspberry pi. We'll use OSOYOO Robot Car for Raspberry Pi to do some simple movements.

If you have passed the test movement of this lesson, it means Raspberry pi, motors, battery, model X board, OSOYOO PWM HAT and wire connections between these parts are all functioning well, and you can move on other sample lessons.








All lessons are based on the frame work of this lesson. Please follow this lesson carefully.


Video Link: <https://youtu.be/KTfYa4QhIhQ>

### PARTS & DEVICES

No.	Picture	Device	Qty.	Accessories	Link
1		Raspberry pi board 2/3/4 (not in package)	1	M2.5 Plastic Screw x4 M2.5 Plastic Nut x2 M2.5 Plastic Pillar x4 M2.5*12 plastic pillars x2	<a href="#">Click here to buy</a>
2		Pie car chassis	1	M3*45 pillars x 4 M3 *10 hex screws x8 M3 washer x 8	<a href="#">Click here to buy</a>

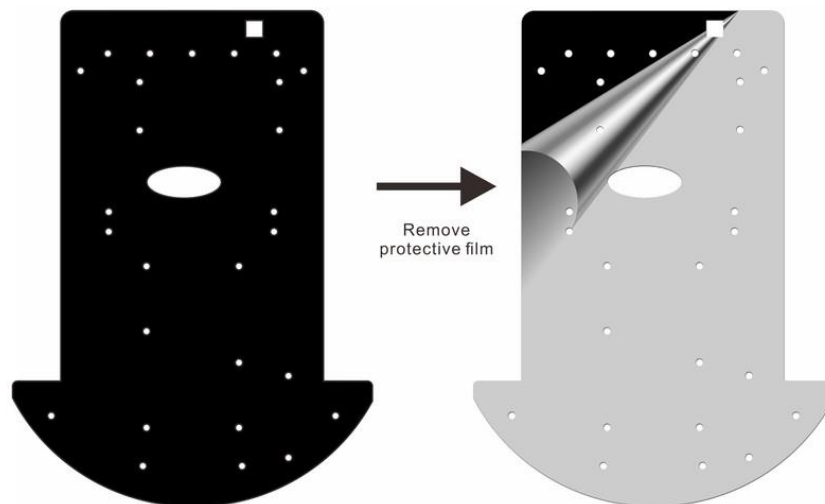
3		Gear motor	2	Motor Holders with screws x2	<a href="#">Click here to buy</a>
4		Wheel	1	screws for wheels x2	<a href="#">Click here to buy</a>
5		Universal Wheel	1	M3 *10 hex screws x2 M3 nuts x2	<a href="#">Click here to buy</a>
6		<a href="#">OSOY00 PWM HAT</a>	1	M2.5*13 plastic pillars x2	<a href="#">Click here to buy</a>
7		OSOY00 model X motor driver module	1	M2.5 Plastic Screw x 4 M2.5 Plastic Nut x4 M2.5 Plastic Pillar x4	<a href="#">Click here to buy</a>
8		Voltage meter	1	M2.5 Plastic Screw x 2 M2.5 Plastic Nut x2 M2.5 Plastic Pillar x 2	<a href="#">Click here to buy</a>
9		18650 battery case	1	M3*10 screw x4 M3 nuts x4	<a href="#">Click here to buy</a>

10		18650 battery charger	1		<a href="#">Click here to buy</a>
11		2000mAh 18650 Rechargeable battery	1		<a href="#">Click here to buy</a>
12		Metal Motor Holders with screws	2		<a href="#">Click here to buy</a>
13		3pin female to 3pin female jumper wire	1		<a href="#">Click here to buy</a>
14		OSOY00 2pin PnP cable 20cm	1		<a href="#">Click here to buy</a>
15		6Pin female to female jumper wire	1		<a href="#">Click here to buy</a>
16		Philips screwdriver	1		<a href="#">Click here to buy</a>

17		Hex Screwdriver	1		<a href="#">Click here to buy</a>
----	---	-----------------	---	--	-----------------------------------

## HARDWARE INSTALLATION

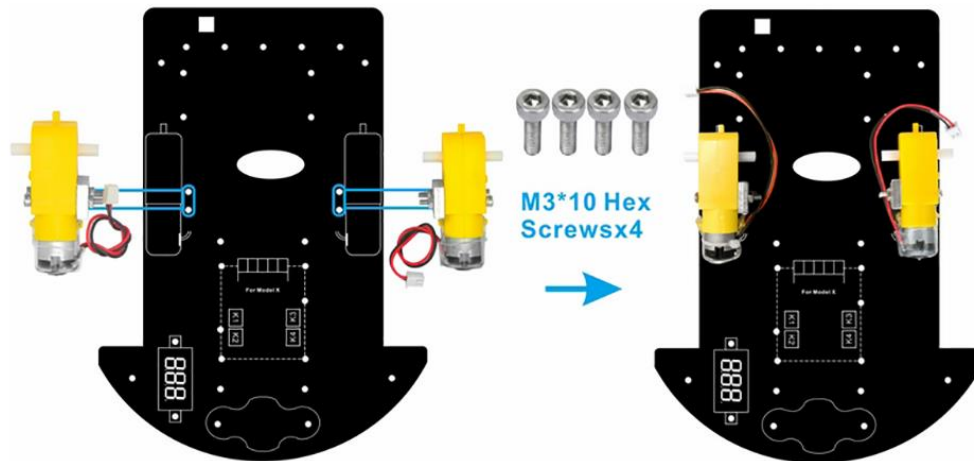
1) Remove the protective film on upper and lower car chassis (Each car chassis has one protective film).



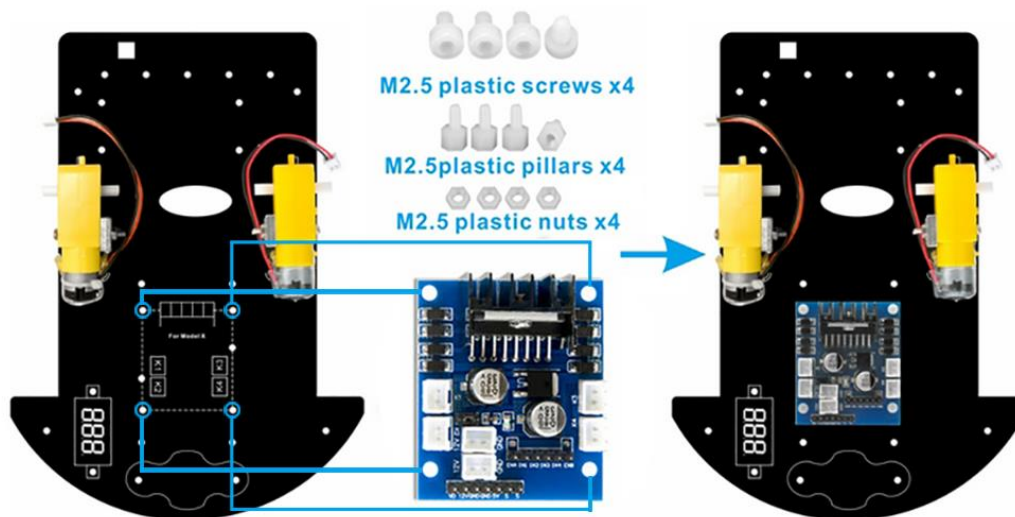
2) Install 2 motors with Metal Motor Holders as follows.



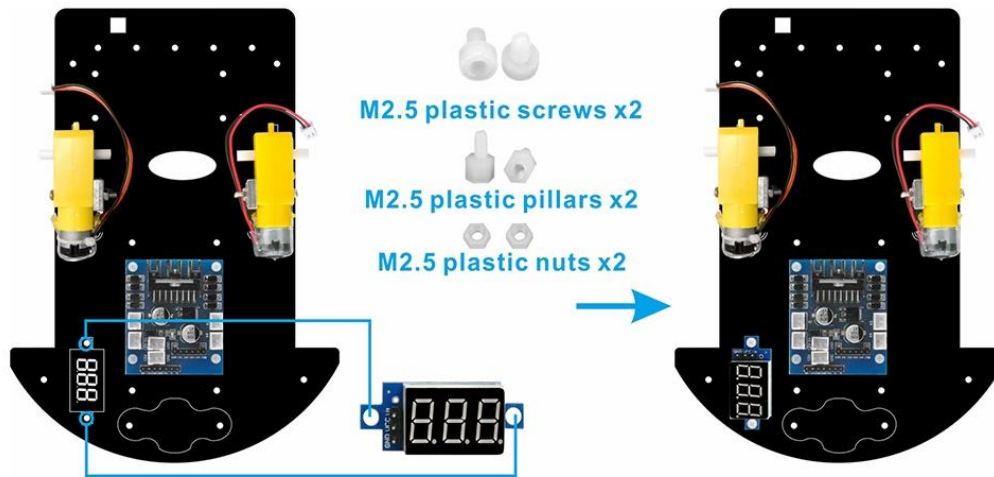
3) Install 2 motors on lower car chassis with screw M3\*10. (Screws in metal motor holder package. Please add M3 washers between screws and chassis if the screws come off easily.)



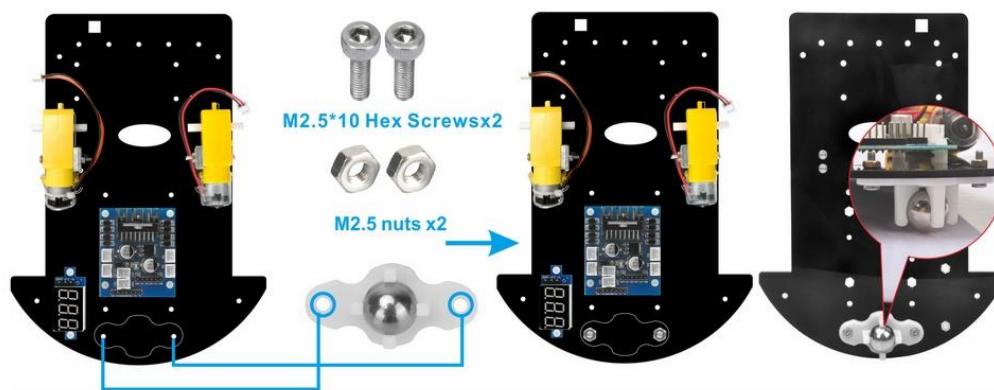
4) Install OSOYOO MODEL X motor driver module to car lower chassis with 4pcs M2.5 plastic screws, plastic pillars and plastic nuts. (Please make sure you install the OSOYOO MODEL X motor driver module in correct direction.)



5) Install voltage meter on lower car chassis with 2pcs M2.5 plastic screws, plastic pillars and plastic nuts.



6) Install universal wheel under the lower car chassis with 2pcs M3x10 hex screws and M3 nuts.



7) Install 2 wheels onto the motors (If you find the wheels install too tightly to move, you can remove the screws on wheels)

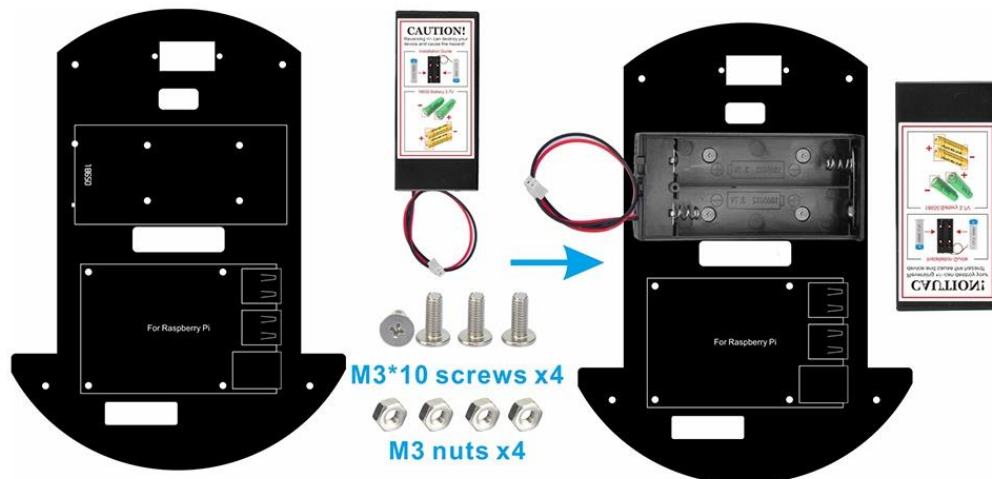




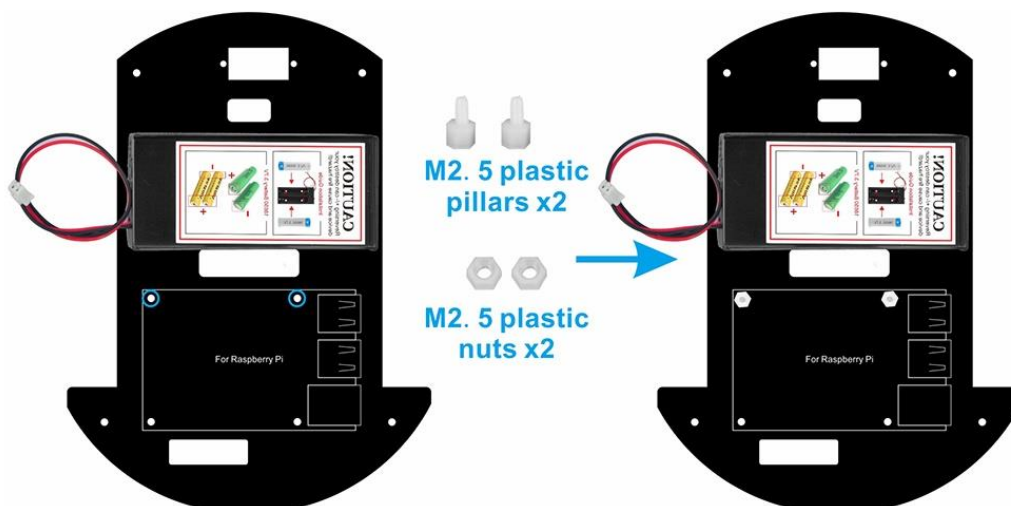
8) Install 4pcs copper pillars on low car chassis with 4pcs M3x10 hex screws. (Please add M3 washes between screws and chassis if the screws come off easily.)



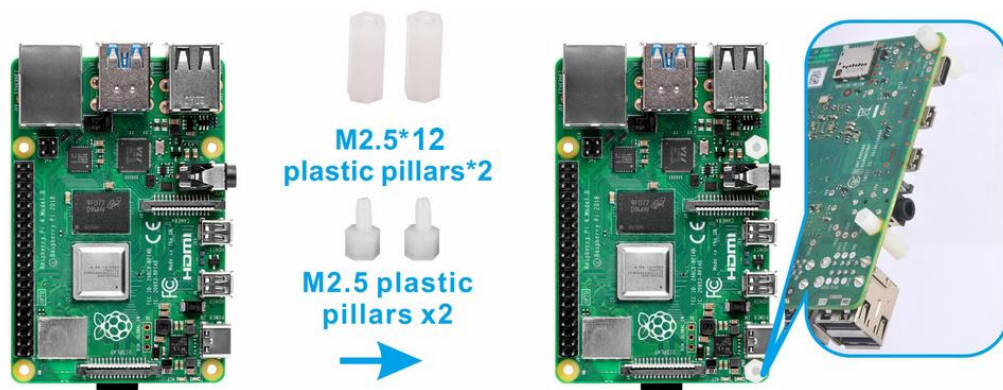
9) Install 18650 battery box on upper car chassis with 4pcs M3x10 screws and M3 nuts.



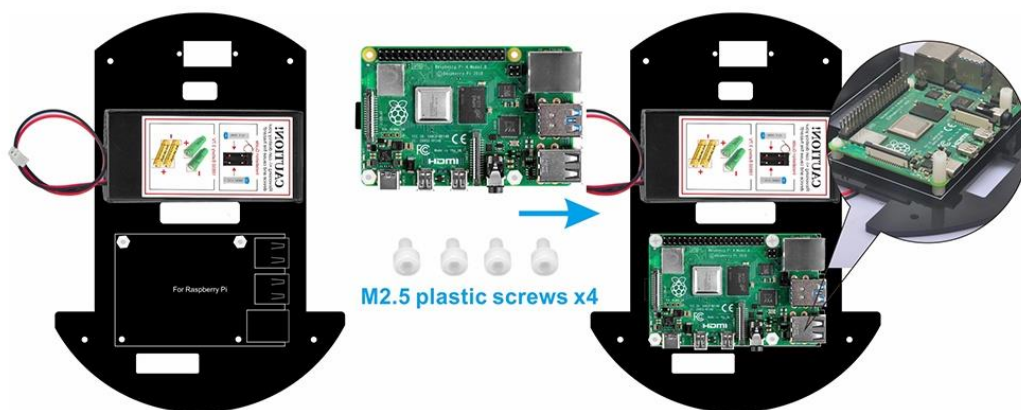
10) Fix 2pcs M2.5 plastic pillars on upper chassis as follow:



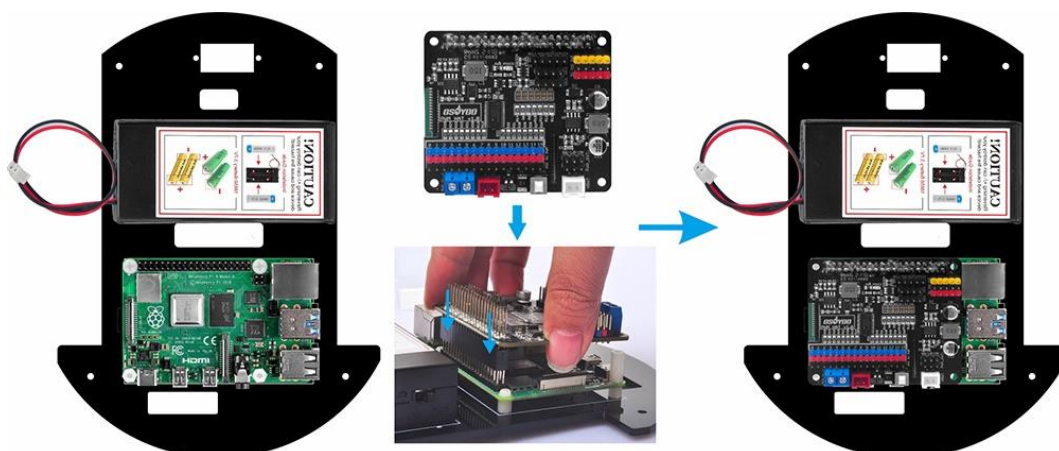
11) Use M2.5 plastic pillar cross Raspberry Pi and M2.5\*12 plastic pillar from bottom to top as follow



12) Install 2pcs M2.5 plastic screws under the chassis and 2pcs M2.5 plastic screws on Raspberry Pi

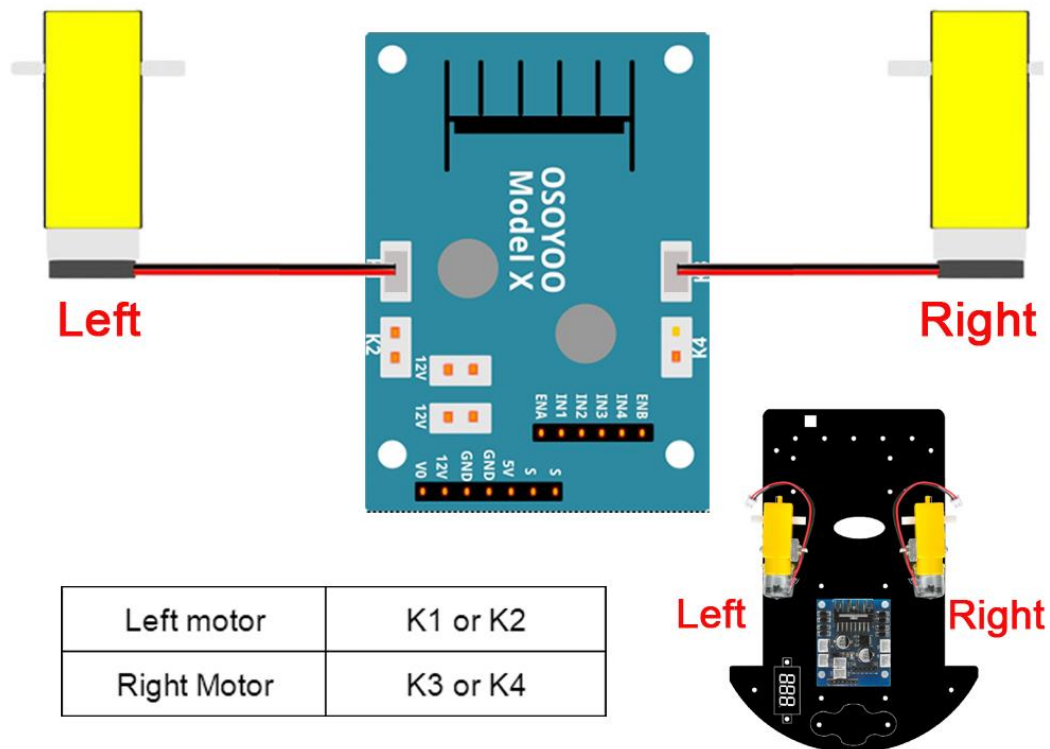


13) Insert OSOYOO PWM Hat on Raspberry Pi as follow:

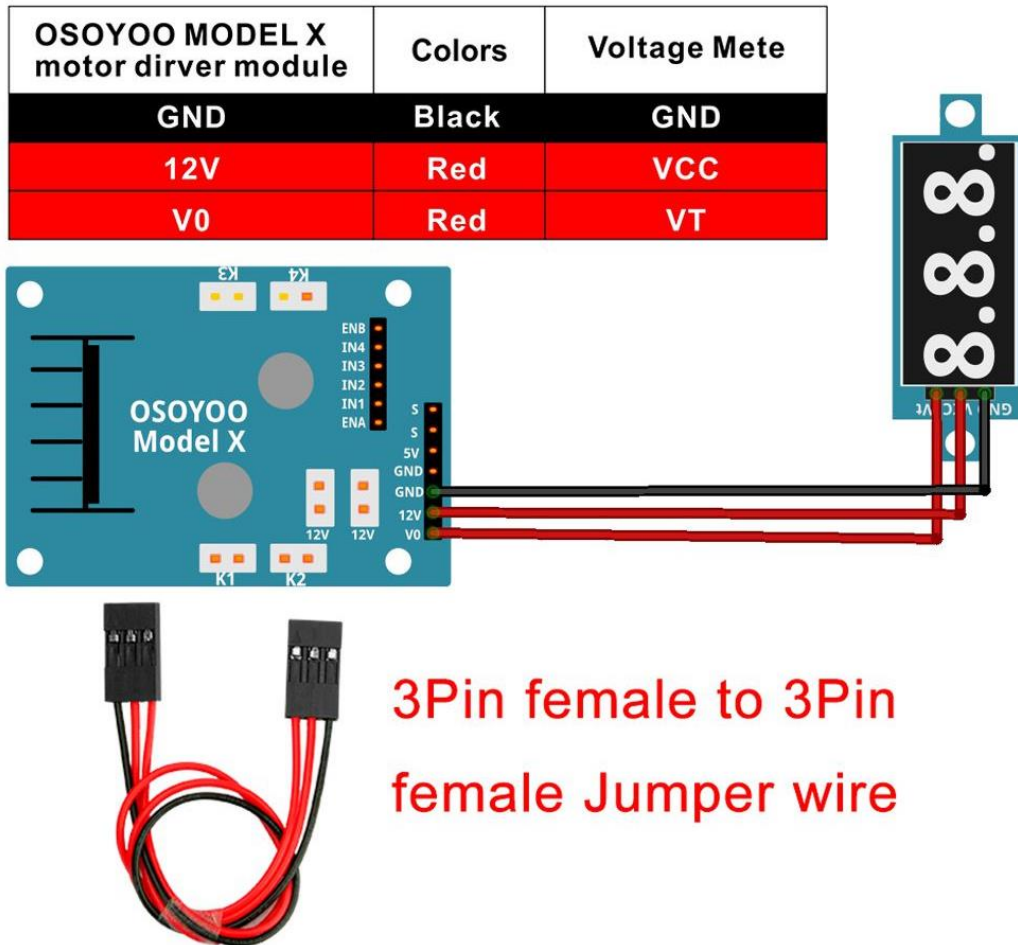


CIRCUIT CONNECTION

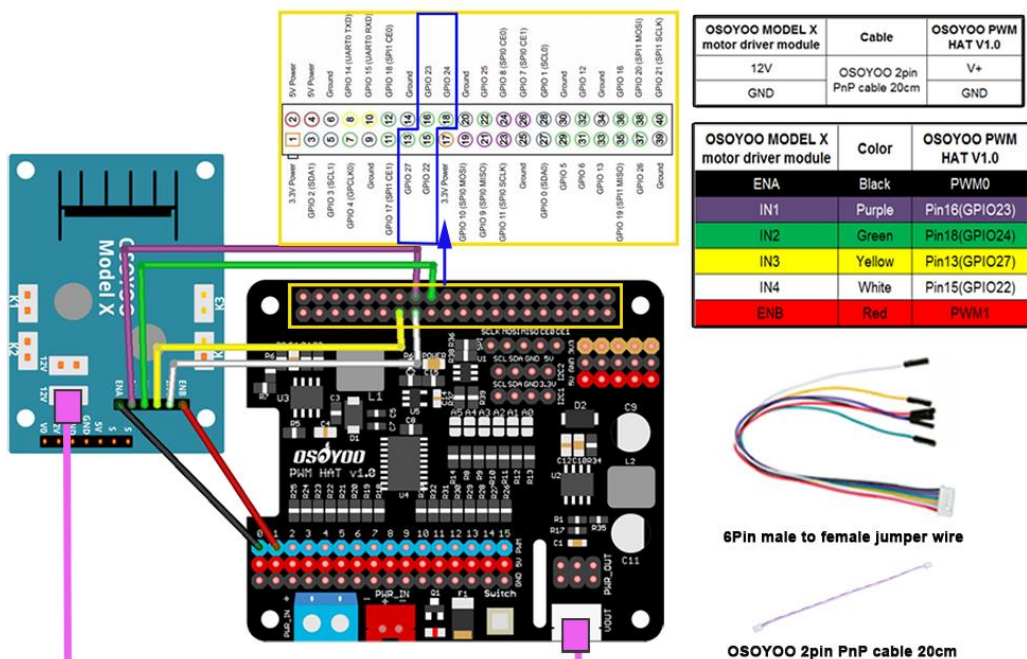
Step 1: CONNECT left motor to K1 or K2 socket of OSOYOO Model X motor driver module , right motor to K3 or K4 socket as per following graph:



Step 2: Connect Voltage Meter to OSOYOO MODEL X motor driver module with 3pin female to female jumper wires as below connection diagram:

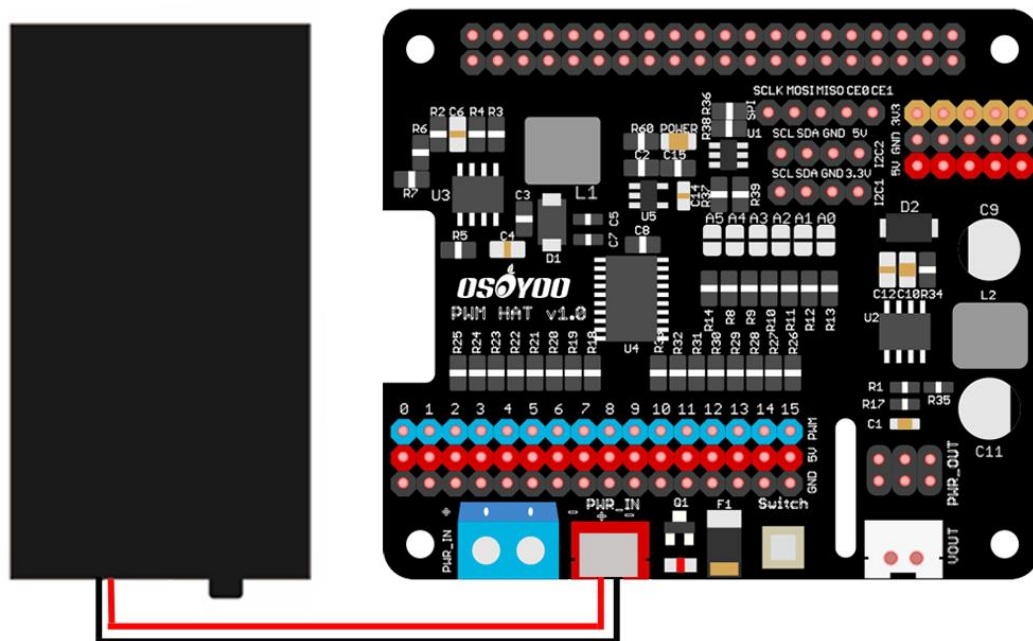


Step 3: Connect OSOYOO MODEL X motor driver module to OSOYOO PWM Hat V1.0 with 6Pin female to female jumper wire, also connect 2pin 20cm cable as following graph:





Step 4: Connect 18650 battery box to OSOY00 PWM Hat V1.0 as below connection diagram:



Step 5: Connect upper chassis to lower chassis with 4pcs M3\*10 hex screws and 4pcs M3 washers



Step 6: Please install your 18650 batteries in battery box for 18650 as per following instruction (**Note: Check the box instruction and make sure polar direction is correct, otherwise it can destroy your device and cause fire hazard.**)

# CAUTION!

Reversing +/- can destroy your device and cause fire hazard

## Installation Guide



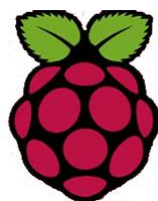
## 18650 Battery 3.7V





Now hardware installation is almost down.

## SOFTWARE INSTALLATION

Raspberry pi OS  
(Raspberry Pi Legacy  
(buster))



Download Raspberry pi OS from:  
[www.raspberrypi.com/software/operating-systems/#raspberry-pi-os-legacy](http://www.raspberrypi.com/software/operating-systems/#raspberry-pi-os-legacy)

Imager utility: Win32DiskImager utility		<a href="#">Download</a> for free
Format Tool: SDformatter (Optional)		<a href="#">Download</a> for free
SSH Tool: PuTTY (for Windows user)		<a href="#">Download</a> for free

Software Preparation:

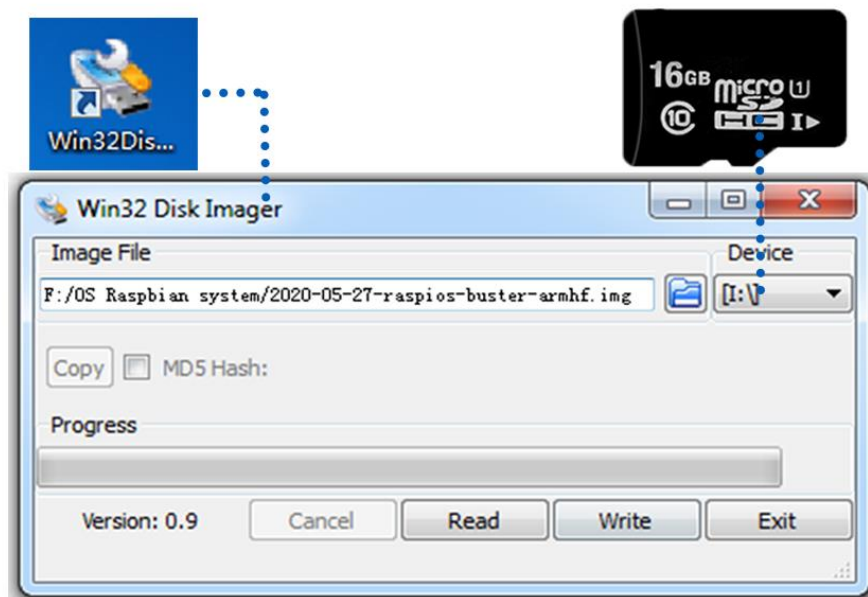
**Operation System for this robot car:** **Raspberry Pi Legacy(buster)** . Please be noted that this robot car sample projects do **NOT** support Raspberry Pi New **Bullseye** OS. So don't make mistake on OS download.

**Here are the steps to install the Raspberry Pi Legacy OS:**

**Step 1: Download Raspberry Pi Buster Operation System(OS) and burn it onto SD card** (skip this step if your SD card has pre-installed Rpi Buster OS Image).

You can select the of Raspberry Pi OS (Legacy) on the official website: <https://www.raspberrypi.com/software/operating-systems/#raspberrypi-os-legacy>. Write the image via **Win32DiskImager utility** into your microSD/TF card(minimum 16G) via SD card reader.

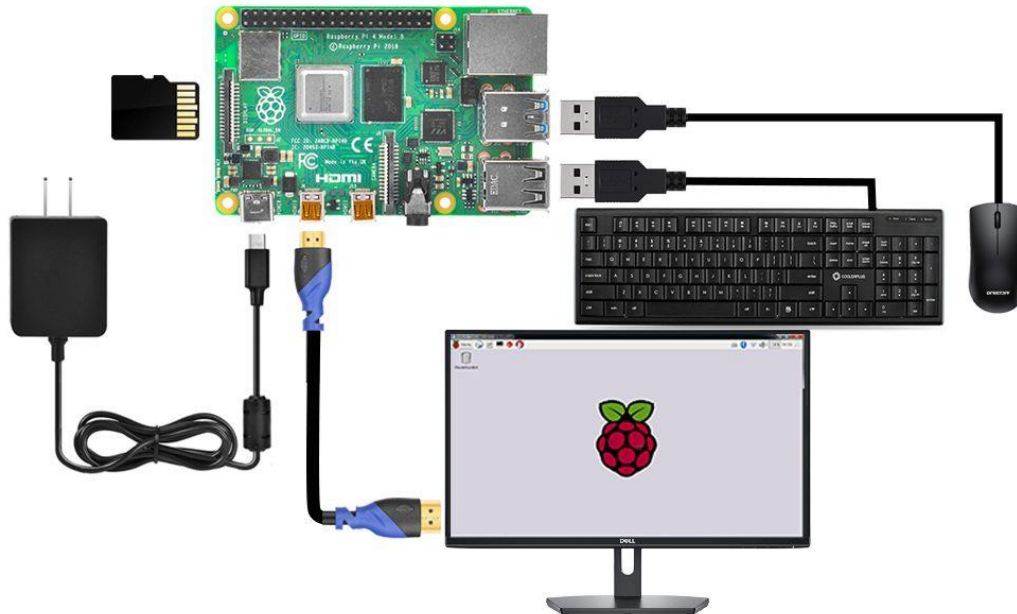
Prepare a new Micro SD card (minimum 16G) and insert it in USB micro SD card reader, and connect USB micro SD card reader with your PC.



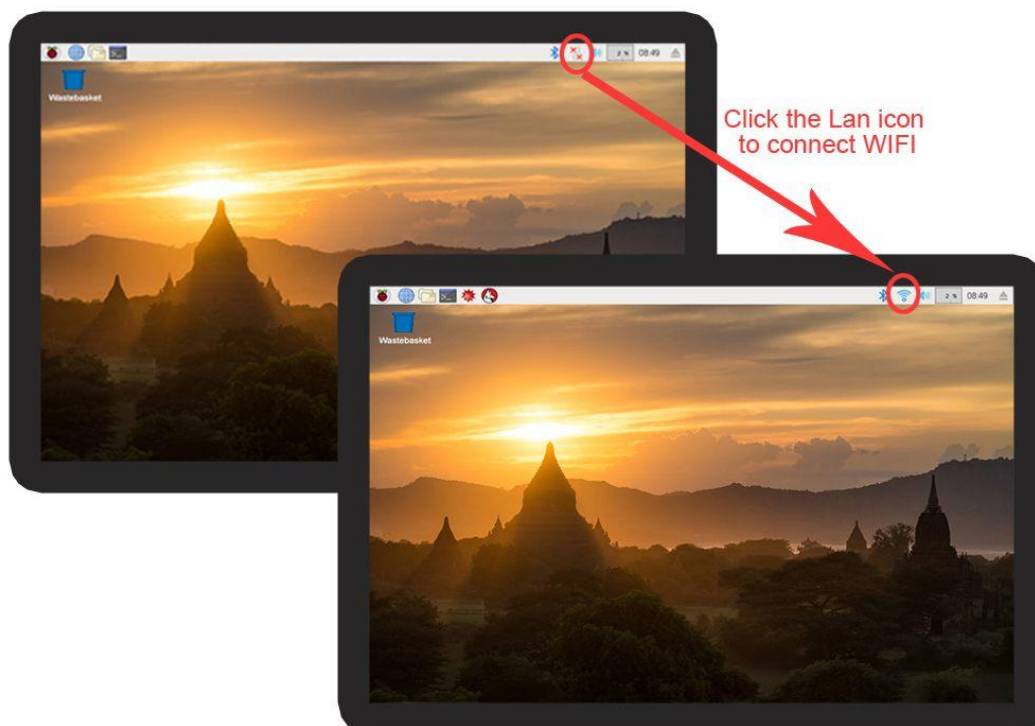
## Step 2: Connect Wifi

1. Connect Raspberry Pi to your HDMI monitor or TV. Put a keyboard and mouse into Raspberry Pi USB ports. Insert SD card(from Step1) into the slot on your Raspberry Pi (if you have intalled your raspberry pi robot car, please replace the raspberry Pi in the following picture with your robot car).

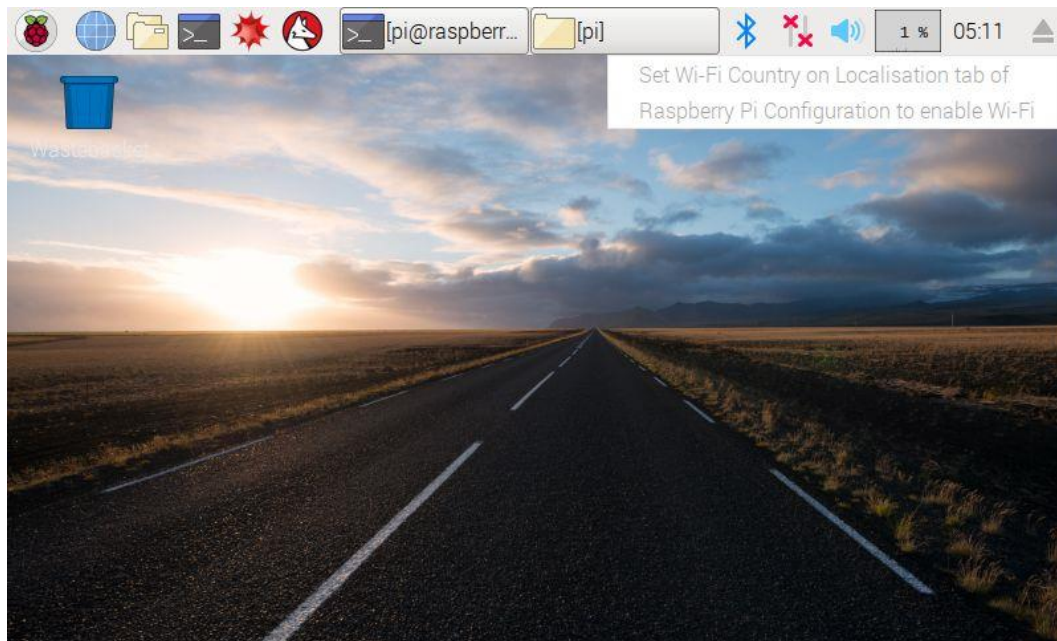




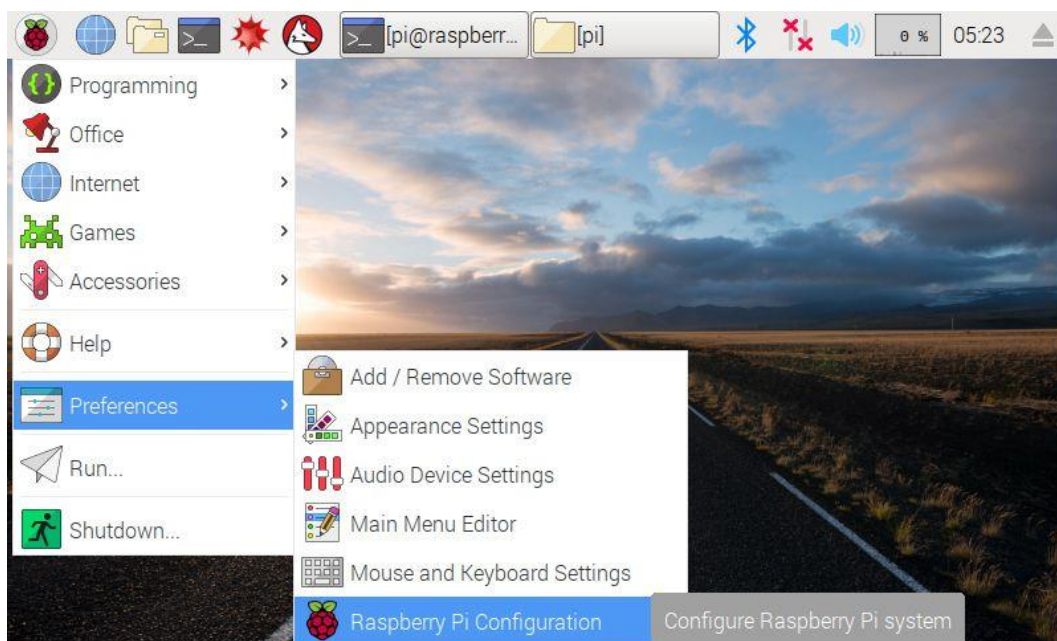
2. Please click the Lan icon at the right-upper corner, and select your WIFI SSID, and enter the password of your wifi to connect wifi hotspot. Then you can see the console full screen.

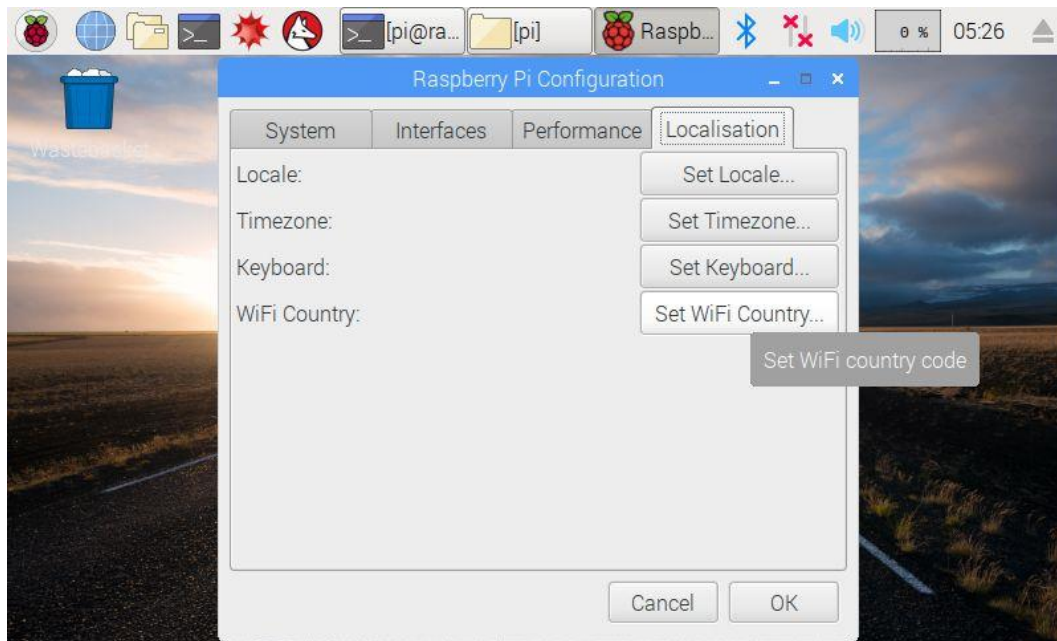


3. If you move over the Lan icon, and get the notice as following:



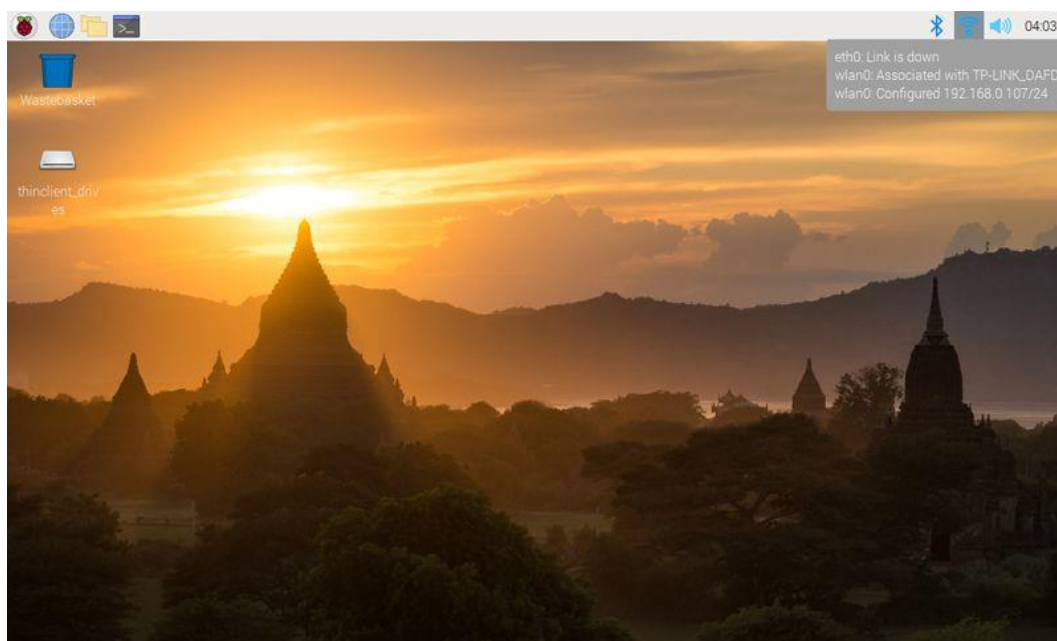
4. Please click configure the WIFI localisation as following:





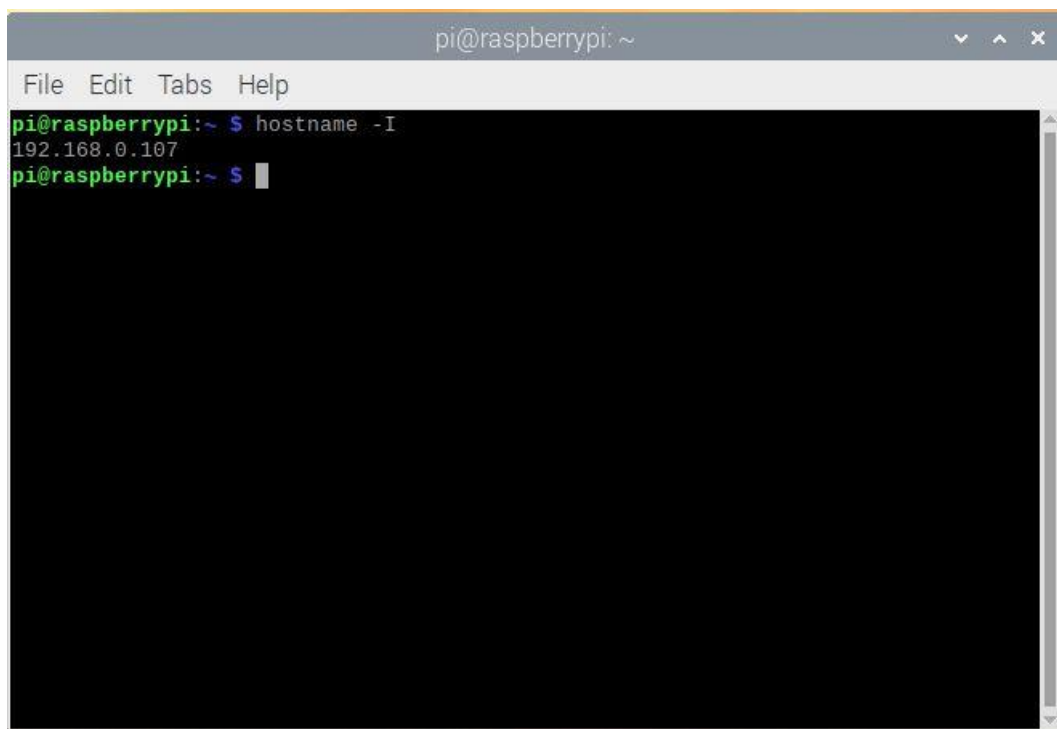
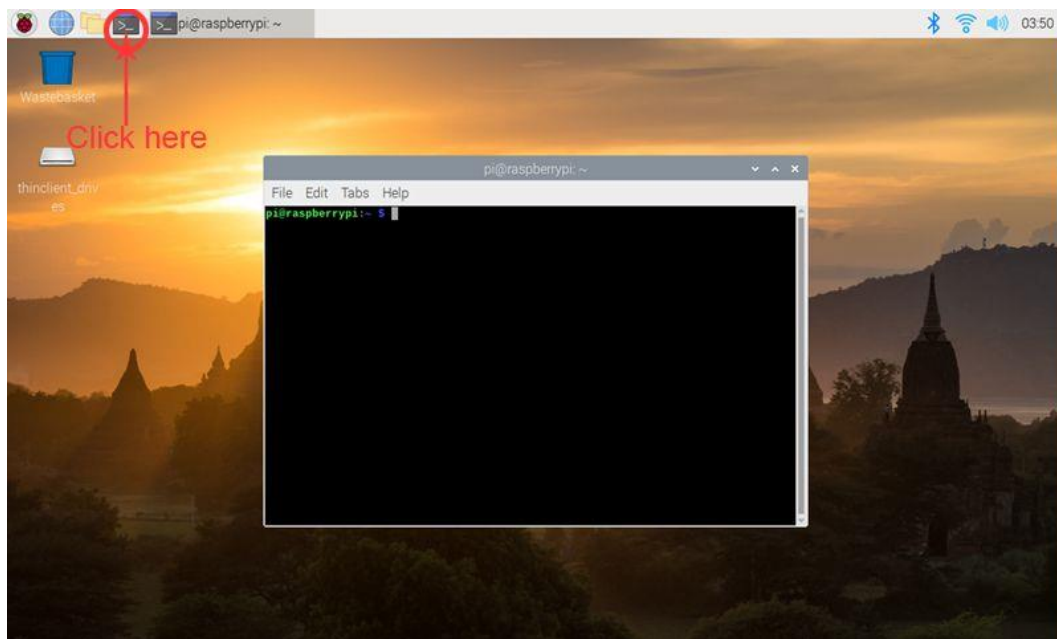
### Step 3. Find out the IP address of the RPi.

**Method A:** Connect your Pi to monitor and mouse, click LAN or WIFI icon to get the IP address as following photo



**Method B:** Click the icon of Terminal on the screen, enter the following command in terminal window to get your IP address :

```
hostname -I
```

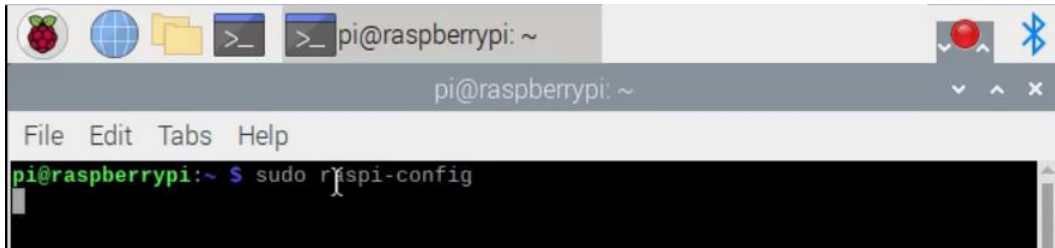


#### Step 4: Enable SSH

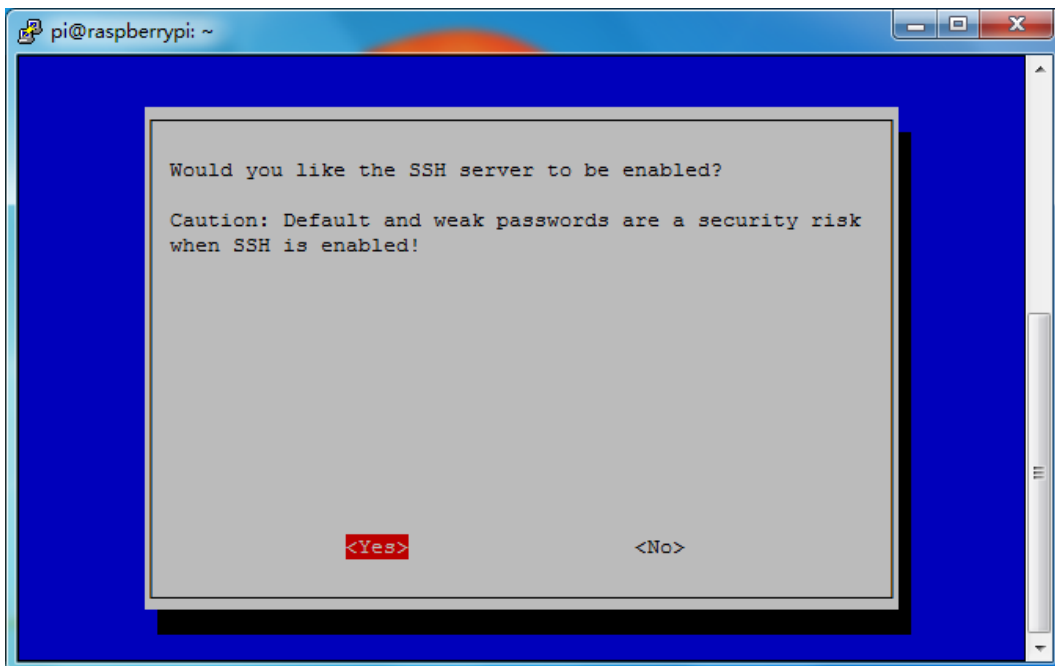
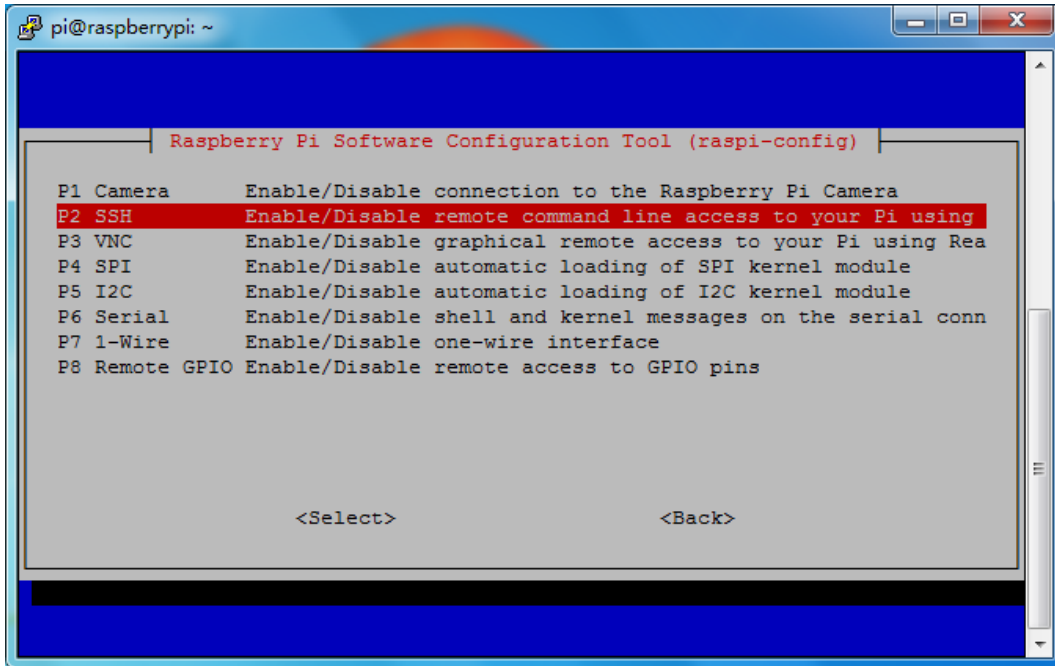
press **Ctrl-T** and open command terminal. Then type :

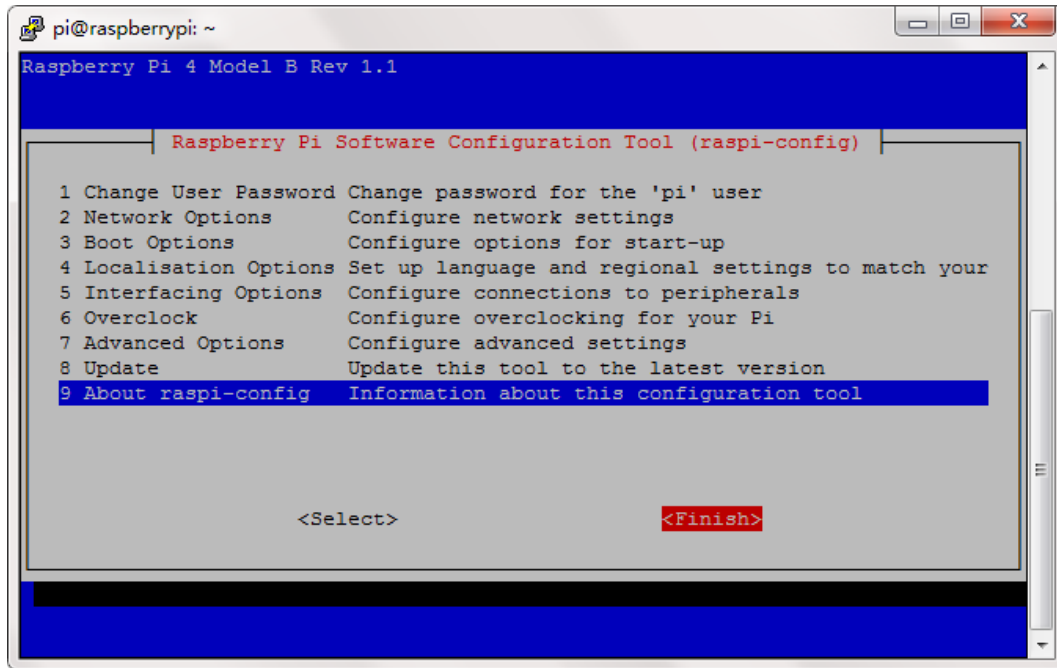
`sudo raspi-config`





You will go to configuration menu, select **Interface Options** ->**SSH** -> **Yes** ->**OK** ->**Finish**



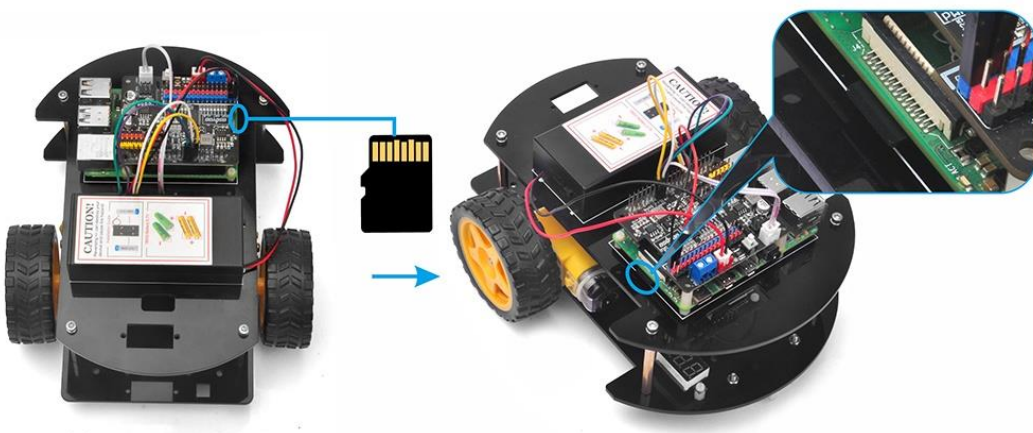


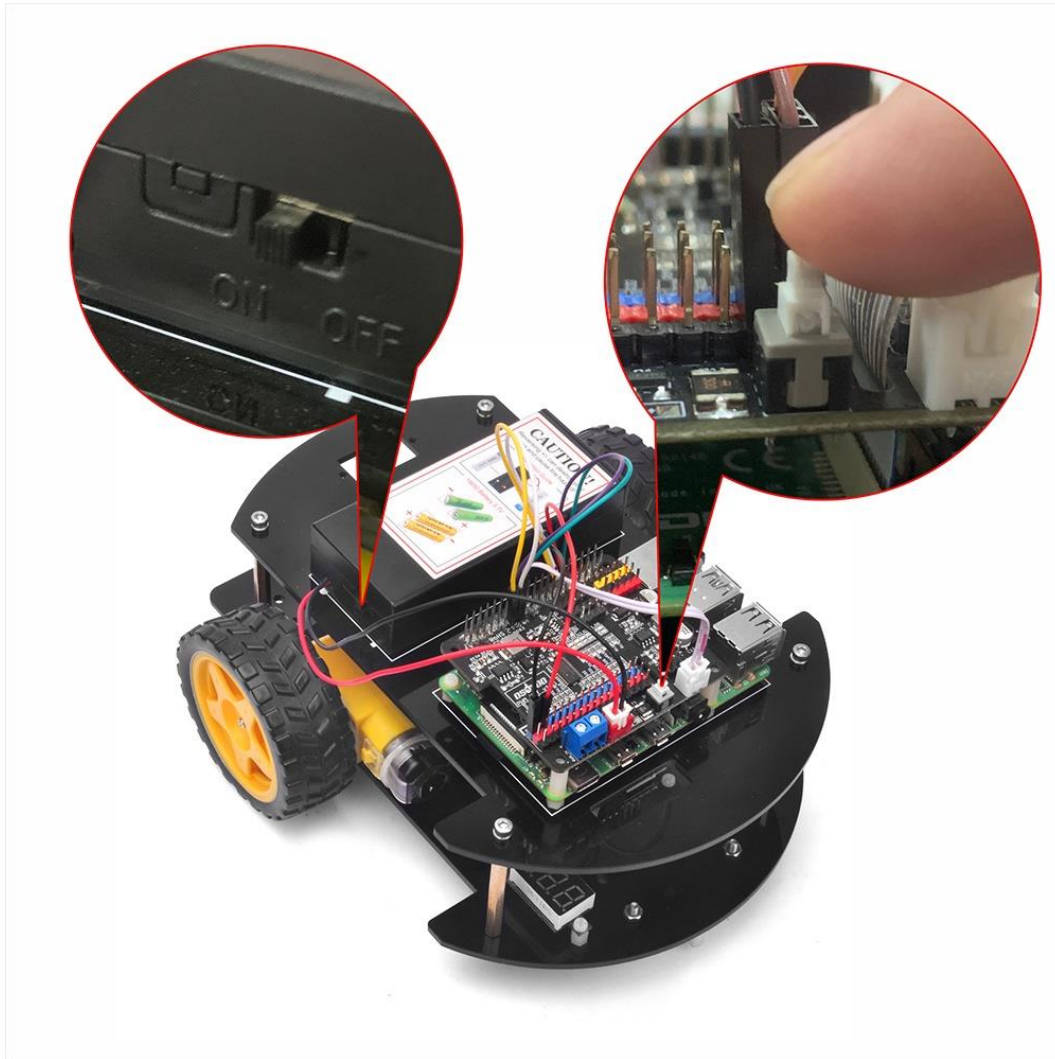
## Step 5: Use ssh tool to control Raspberry Pi remotely

### Note:

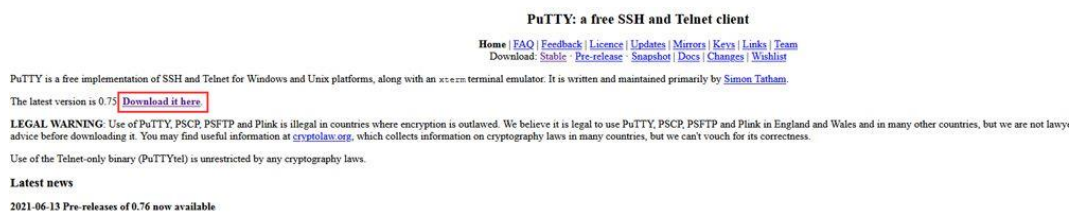
- 1) To learn more about how to use ssh tool to control Raspberry Pi remotely, Please visit the link: <https://osoyoo.com/2017/06/20/raspberry-pi-3-basic-tutorial/#7>
- 2) In our sample lessons, we use Windows PC as the remote console device and PuTTY as the ssh tool

1. Please install the Raspberry Pi with SD card (from step 4) on the robot car (**Attention** please: you need to power off the raspberry pi firstly), and then power on robot car.

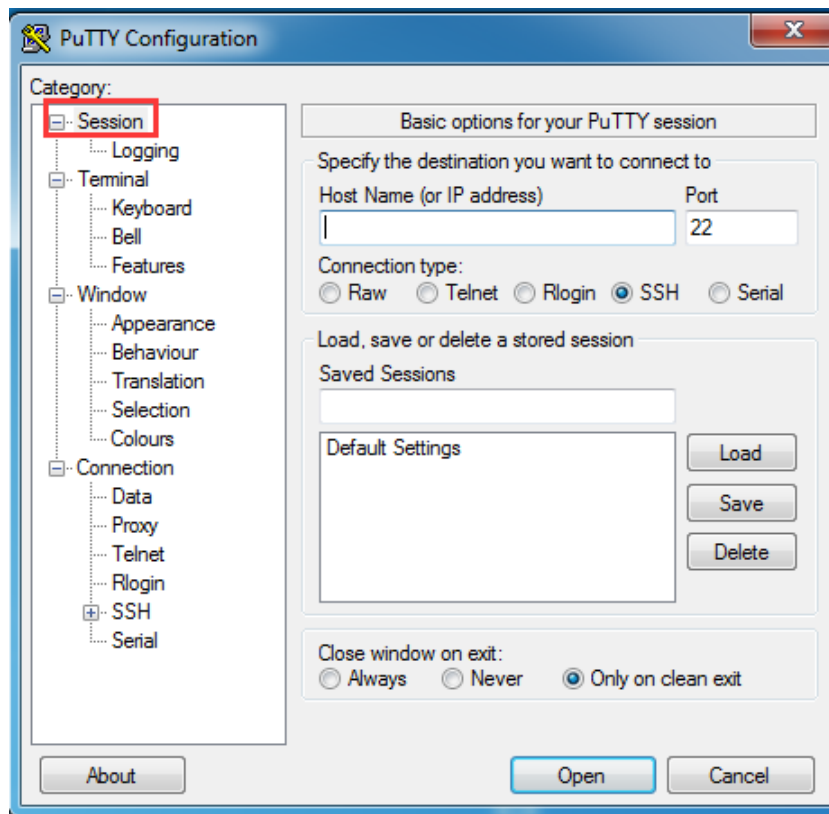




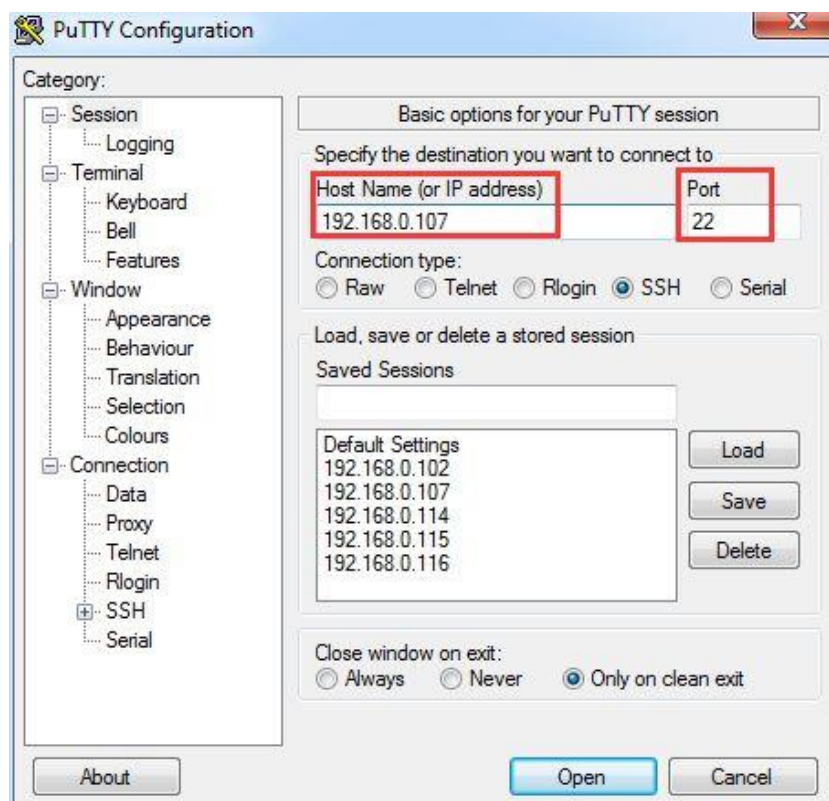
2. Download PuTTY from: <https://www.chiark.greenend.org.uk/~sgtatham/putty/> and install this exe. in your Windows PC



3. Open PuTTY and click Session on the left tree-alike structure (generally it's collapsed upon PuTTY startup):



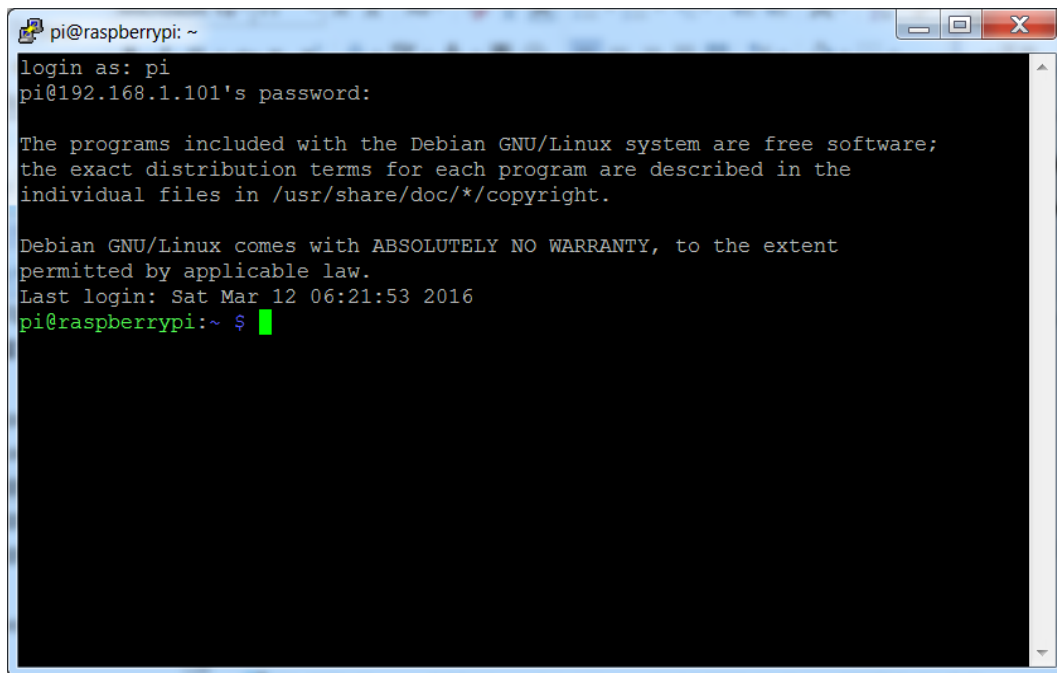
4. Enter the IP address you got into the textbox under Host Name (or IP address) and 22 under Port (by default it is 22) ,then click open.





5. Note that when you first log in to the Raspberry Pi with the IP address, you'll be prompted with a security reminder. Just click **Yes**. When the PuTTY window prompts login as: type in the user name: **pi**, and password: **raspberry** (the default one, if you haven't changed it).

Note: when you're typing the password in, the window shows nothing just null, but you're in fact is typing things in. So just focus on typing it right and press Enter. After you log in the RPi successfully, the window will display as follows:



```
pi@raspberrypi: ~
login as: pi
pi@192.168.1.101's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Mar 12 06:21:53 2016
pi@raspberrypi:~ $
```

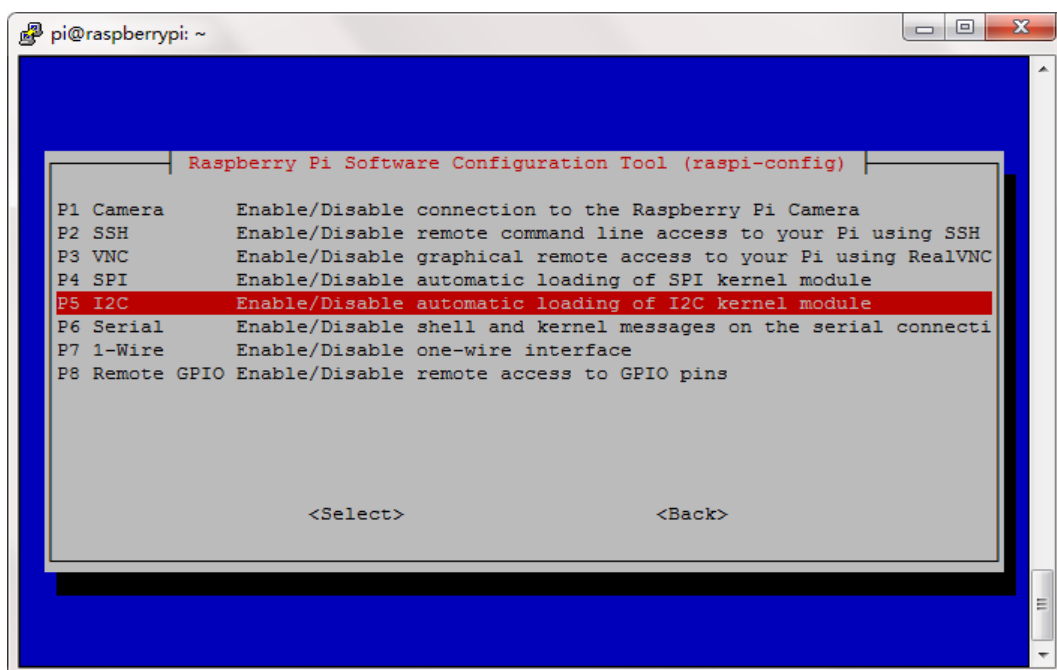
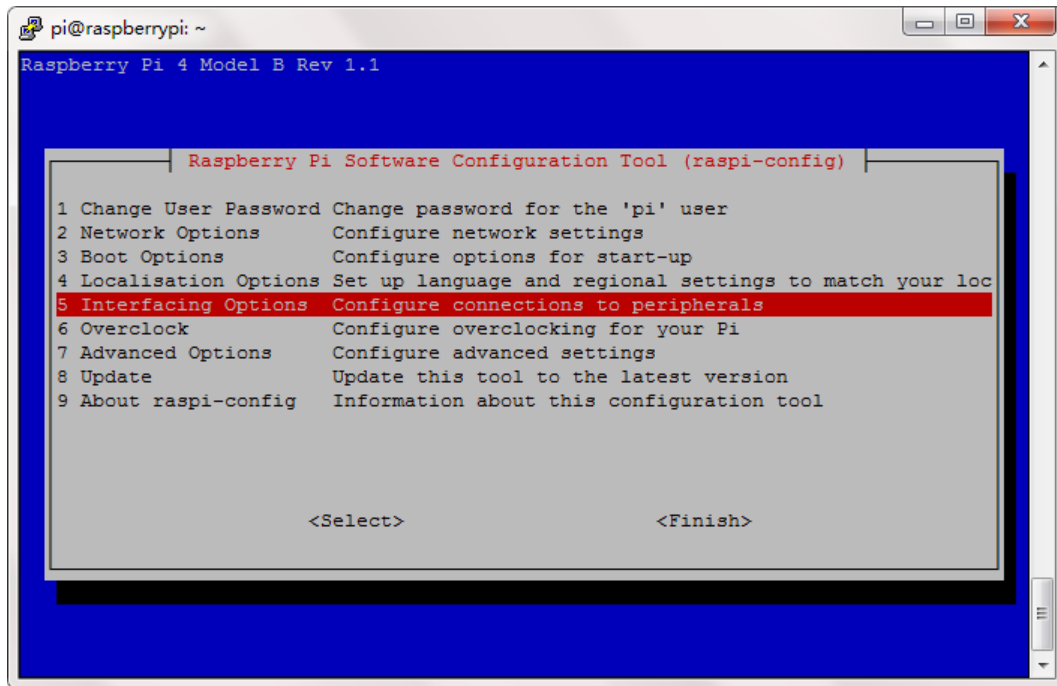
Use ssh tool to control Raspberry Pi remotely on other OS PC, Please visit the link: <https://osoyoo.com/2017/06/20/raspberry-pi-3-basic-tutorial/#7>

## Step 6: Enable I2C

If you have not enable I2C, please run following command to enable I2C.

```
sudo raspi-config
```

Then select Interfacing Options->I2C->Yes->Ok->Finish



**Step 7: Install GPIO Library by running the following three commands:**

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install rpi.gpio
```

```
pi@raspberrypi:~$ sudo apt-get update
Get:1 http://archive.raspberrypi.org/debian buster InRelease [32.6 kB]
Get:2 http://archive.raspberrypi.org/debian buster/main armhf Packages [400 kB]
Get:3 http://raspbian.raspberrypi.org/raspbian buster InRelease [15.0 kB]
Get:4 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13.0 MB]
Get:5 http://raspbian.raspberrypi.org/raspbian buster/non-free armhf Packages [10 kB]
Fetched 13.6 MB in 21s (657 kB/s)
Reading package lists... Done
pi@raspberrypi:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  curl firmware-atheros firmware-brcm80211 firmware-libertas
  firmware-misc-nonfree firmware-realtek ghostscript git git-man
  libcurl3-gnutls libcurl4 libde265-0 libgs9 libgs9-common libmariadb3
  libopenexr23 libpam-systemd libpcre2-16-0 libpcre2-8-0 libpcre2-posix0
  libsystemd0 libtiff5 libudev1 libudisks2-0 mariadb-common
  python-cryptography python-werkzeug python3-cryptography python3-werkzeug
  systemd systemd-sysv tzdata udev udisks2 wireless-regdb xserver-common
  xserver-xorg-core
37 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 89.2 MB of archives.
After this operation, 100 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.raspberrypi.org/debian buster/main armhf xserver-common all
2:1.20.4-1+rpt4+deb10u9 [2,236 kB]
```

```
pi@raspberrypi:~$ sudo apt-get install rpi.gpio
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'python-rpi.gpio-dbgSYM' for regex 'rpi.gpio'
Note, selecting 'python3-rpi.gpio' for regex 'rpi.gpio'
Note, selecting 'rpi.gpio-common' for regex 'rpi.gpio'
Note, selecting 'python-rpi.gpio' for regex 'rpi.gpio'
Note, selecting 'python3-rpi.gpio-dbgSYM' for regex 'rpi.gpio'
python-rpi.gpio is already the newest version (0.7.0-0.1~bpo10+4).
python3-rpi.gpio is already the newest version (0.7.0-0.1~bpo10+4).
rpi.gpio-common is already the newest version (0.7.0-0.1~bpo10+4).
rpi.gpio-common set to manually installed.
The following NEW packages will be installed:
  python-rpi.gpio-dbgSYM python3-rpi.gpio-dbgSYM
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 88.2 kB of archives.
After this operation, 126 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.raspberrypi.org/debian buster/main armhf python-rpi.gpio-dbgSYM armhf 0.7.0-0.1~bpo10+4 [42.7 kB]
Get:2 http://archive.raspberrypi.org/debian buster/main armhf python3-rpi.gpio-dbgSYM armhf 0.7.0-0.1~bpo10+4 [45.5 kB]
Fetched 88.2 kB in 2s (55.6 kB/s)
Selecting previously unselected package python-rpi.gpio-dbgSYM.
(Reading database ... 99385 files and directories currently installed.)
Preparing to unpack .../python-rpi.gpio-dbgSYM_0.7.0-0.1~bpo10+4_armhf.deb ...
Unpacking python-rpi.gpio-dbgSYM (0.7.0-0.1~bpo10+4) ...
Selecting previously unselected package python3-rpi.gpio-dbgSYM.
```

**Step 8: Install pca9685 PWM Library by running the following three commands**

**(ATTENTION: please select a correct command according to your Pi board):**

If you are using Python 2 in Raspberry Pi 3, type:

```
sudo pip install adafruit-circuitpython-pca9685
sudo pip install adafruit-circuitpython-servokit
```

If you are using Python 3 in Raspberry Pi 4/3A+, type:

```
sudo pip3 install adafruit-circuitpython-pca9685
```

```
sudo pip3 install adafruit-circuitpython-servokit
```

```
pi@raspberrypi:~ $ sudo pip3 install adafruit-circuitpython-pca9685
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting adafruit-circuitpython-pca9685
  Downloading https://files.pythonhosted.org/packages/7a/de/c6af4bba7b2a331fa8c5
6a47f7e6c47dd7dd0703d5e414f29a795fda7f9d/adafruit_circuitpython_pca9685-3.4.10-p
y3-none-any.whl
Collecting adafruit-circuitpython-busdevice (from adafruit-circuitpython-pca9685
)
  Downloading https://files.pythonhosted.org/packages/46/59/1a187155643f491b6bea
2e66d455c8f34751353042e5113a609009c6d257/adafruit_circuitpython_busdevice-5.2.6-
py3-none-any.whl
Collecting adafruit-circuitpython-register (from adafruit-circuitpython-pca9685)
  Downloading https://files.pythonhosted.org/packages/19/82/d2aa20911129f0c57ff5
a02ae64659d563e6dee27f9a4a2e3592ebe44fe3/adafruit_circuitpython_register-1.9.16-
py3-none-any.whl
Collecting Adafruit-Blinka (from adafruit-circuitpython-pca9685)
  Downloading https://files.pythonhosted.org/packages/f3/fd/0331cfff7649c58caac5
afb528b5757cd94edc35744f19430f32b1c1416a/Adafruit_Blinka-8.20.1-py3-none-any.whl
(303kB)
 3% |██████████| 10kB 14kB/s eta 0:00:20
pi@raspberrypi:~ $ sudo pip3 install adafruit-circuitpython-servokit
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting adafruit-circuitpython-servokit
  Downloading https://files.pythonhosted.org/packages/14/34/1a22ee0ea37bcce532d8
372e2303cbbdfalbe01678212a8379ec23ace029/adafruit_circuitpython_servokit-1.3.15-
py3-none-any.whl
Collecting adafruit-circuitpython-busdevice (from adafruit-circuitpython-servoki
t)
  Using cached https://files.pythonhosted.org/packages/46/59/1a187155643f491b6be
a2e66d455c8f34751353042e5113a609009c6d257/adafruit_circuitpython_busdevice-5.2.6
-py3-none-any.whl
Collecting adafruit-circuitpython-register (from adafruit-circuitpython-servokit
)
```

**Step 9: Type following command to download the sample code.**

```
wget http://osoyoo.com/driver/picar/picar-basic.py
```

```
pi@raspberrypi:~ $ wget http://osoyoo.com/driver/picar/picar-basic.py
--2023-04-19 04:27:56-- http://osoyoo.com/driver/picar/picar-basic.py
Resolving osoyoo.com (osoyoo.com)... 2606:4700:3031::6815:2a5e, 2606:4700:3036::
ac43:a078, 172.67.160.120, ...
Connecting to osoyoo.com (osoyoo.com)|2606:4700:3031::6815:2a5e|:80... connected
.
HTTP request sent, awaiting response... 200 OK
Length: 4720 (4.6K) [text/x-python]
Saving to: 'picar-basic.py'

picar-basic.py      100%[=====>] 4.61K --.-KB/s in 0.001s
2023-04-19 04:27:57 (4.67 MB/s) - 'picar-basic.py' saved [4720/4720]
```

**Step 10: Put the robot car on the ground and then run the basic movement python program by typing following command in terminal (**ATTENTION: please select a correct command according to your Pi board**):**



If you are using Python 2 in Raspberry Pi 3/3B+ type:

```
python picar-basic.py
```

If you are using Python 3 in Raspberry Pi 4/3A+ type:

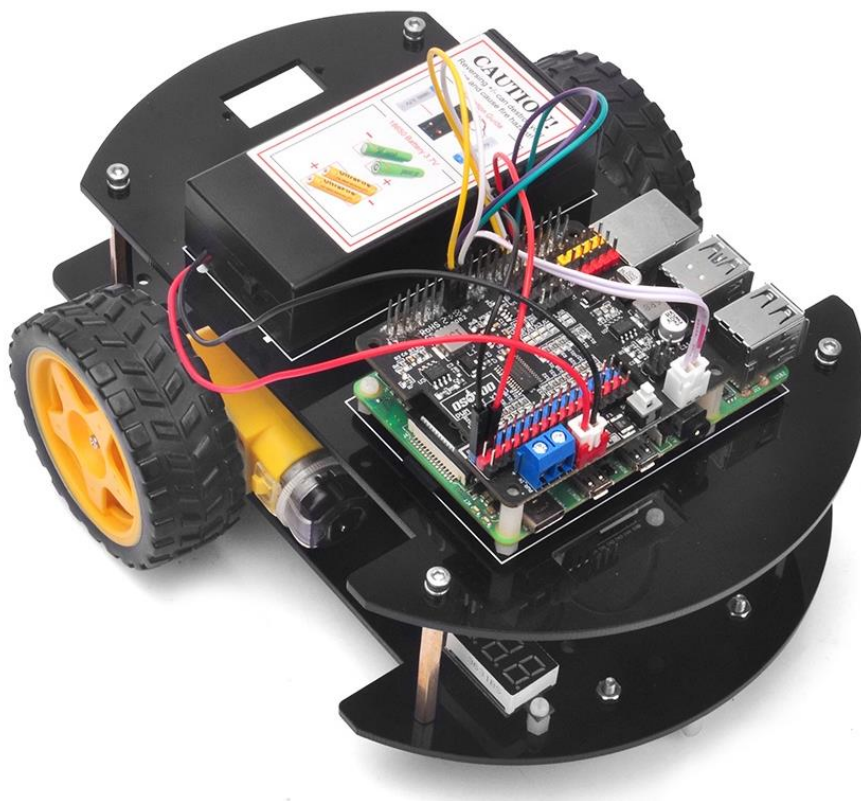
```
python3 picar-basic.py
```

```
pi@raspberrypi:~ $ python3 picar-basic.py  
Moving servo on channel 0, press Ctrl-C to quit...
```

After above python is running, your robot car will move forward for 1 seconds and then move backward for 1 seconds, and turn left for 1 seconds and turn right for 1 seconds then stop. (If the robot car don't move as above, please check your connections firstly)

We have write full comments in the sample python

code <http://osoyoo.com/driver/picar/picar-basic.py> . If you have some basic python knowledge, you can easily understand how to customize the code for your own application.



# OSOY00 Robot car for Raspberry Pi

## Lesson 2: Line Tracking (Python)

### OBJECTIVE




In this lesson, we use Python to program our Raspberry Pi robot car to automatically drive along a black line in white ground. We will use five IR tracking sensors to detect the line.

You will learn how Python read digital data from GPIO pins.

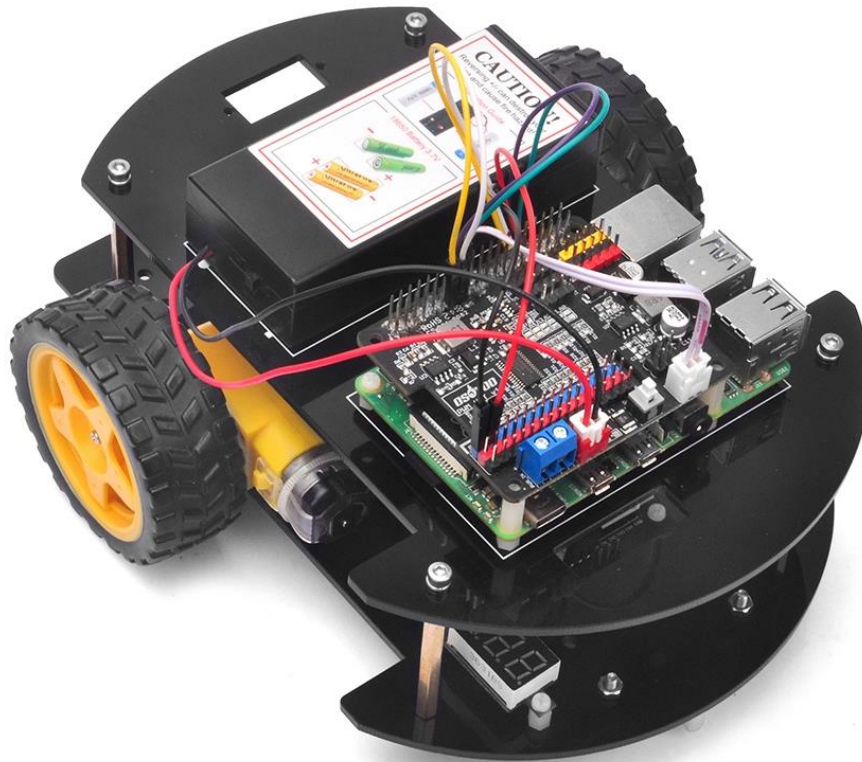
**Video link:** [https://youtu.be/tj3wr0CIR\\_A](https://youtu.be/tj3wr0CIR_A)

### PARTS & DEVICES

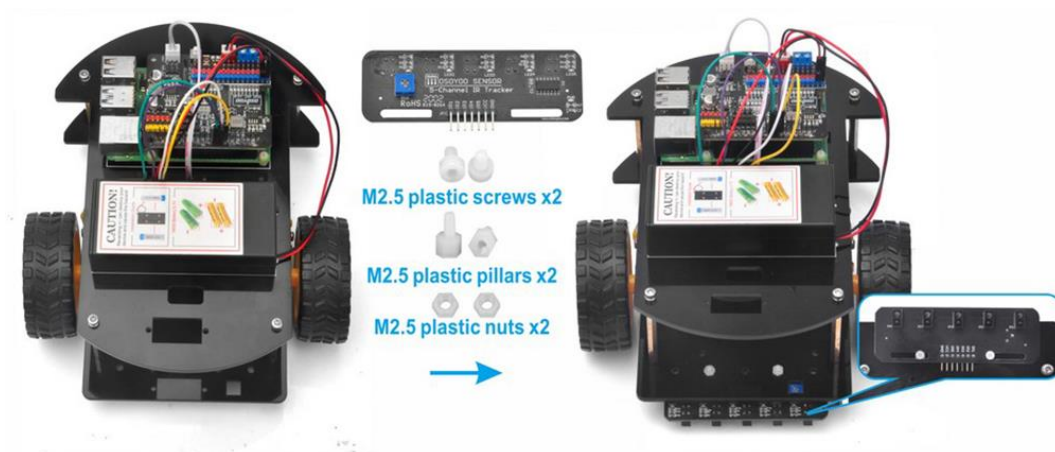
### HARDWARE INSTALLATION

No.	Picture	Device	Qty.	Accessories	Link
1		Tracking sensor module	1	M2.5 Plastic Screw x 2 M2.5 Plastic Nut x2 M2.5 Plastic Pillar x 2	<a href="#">Click here to buy</a>
2		7pin 25 cm Female to Female Cable	1		<a href="#">Click here to buy</a>
3		Philips screwdriver	1		<a href="#">Click here to buy</a>

**Step 1:** You must complete [Lesson 1 basic framework](#).

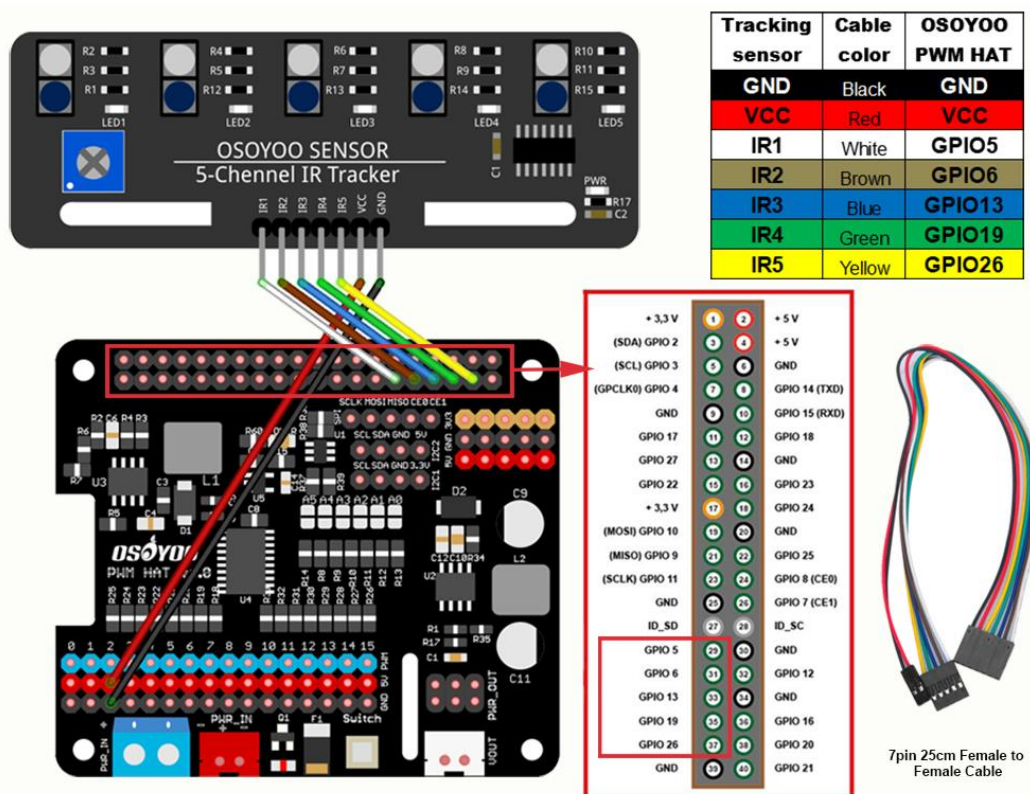


**Step 2:** Install tracking sensor modules under lower car chassis with 2pcs M2.5 plastic screws, M2.5 plastic pillars and M2.5 plastic nuts.



## CIRCUIT CONNECTION

Step 1: Connect GND-VCC pin of tracking sensor module to OSOY00 PWM HAT with 7pin 25 cm female to female cable, as the following photo shows (Remember : DO NOT remove any existing wires installed in Lesson 1) :



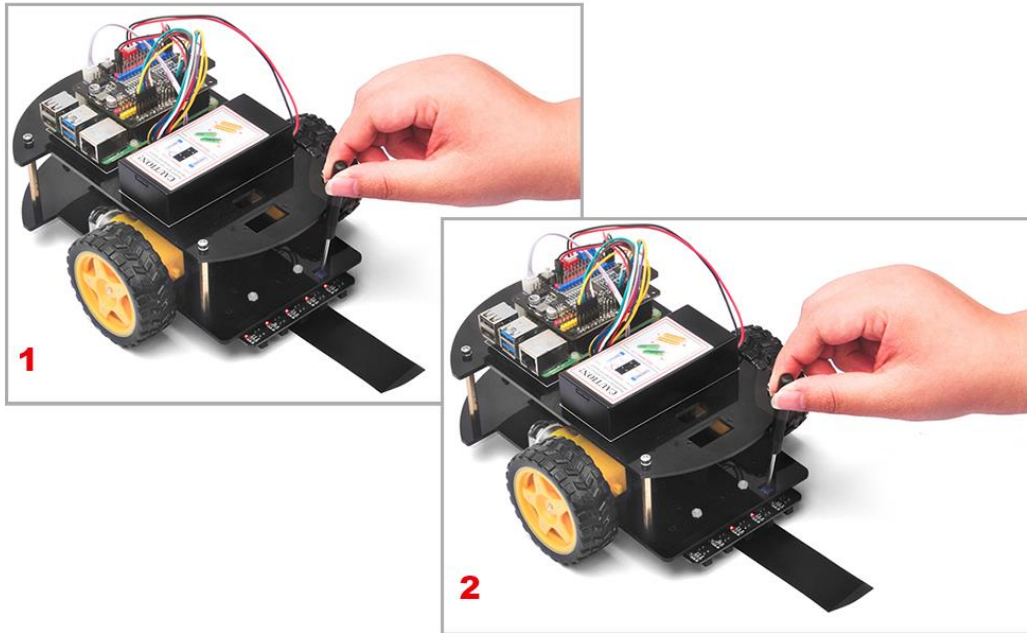
Step 2: Adjust the sensitivity of tracking sensor modules.

Turn on the car, and prepare a black track line on white ground and turn on the car.

Put the car on the black track and white ground repeatedly to adjust the sensitivity of the tracking module.

Please adjust the potentiometer on the tracking sensor and make sure the corresponding indicator LED will be on when the tracking sensor is over black, and the corresponding indicator LED will be off when the tracking sensor is over white.





## SOFTWARE INSTALLATION

Step 1: Make sure you have installed `rpi.gpio`, `adafruit-pca9685` library and enable I2C in [lesson 1](#).

Step 2: Turn on the robot car, then download the python code by typing the following command in your Raspberry Pi terminal:

`wget http://osoyoo.com/driver/picar/picar-line-tracking.py`

```
pi@raspberrypi:~$ wget http://osoyoo.com/driver/picar/picar-line-tracking.py
--2023-04-19 04:49:54-- http://osoyoo.com/driver/picar/picar-line-tracking.py
Resolving osoyoo.com (osoyoo.com)... 2606:4700:3031::6815:2a5e, 2606:4700:3036::a
c43:a078, 172.67.160.120, ...
Connecting to osoyoo.com (osoyoo.com)[2606:4700:3031::6815:2a5e]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5003 (4.9K) [text/x-python]
Saving to: 'picar-line-tracking.py'

picar-line-tracking. 100%[=====>] 4.89K --.-KB/s in 0.002s
2023-04-19 04:49:55 (2.96 MB/s) - 'picar-line-tracking.py' saved [5003/5003]
```

Step 3. Put the car on the black track line, then run the line tracking python program (**ATTENTION: please select a correct command according to your Pi board**),

If you are using Python 2 in Raspberry Pi 3, type:

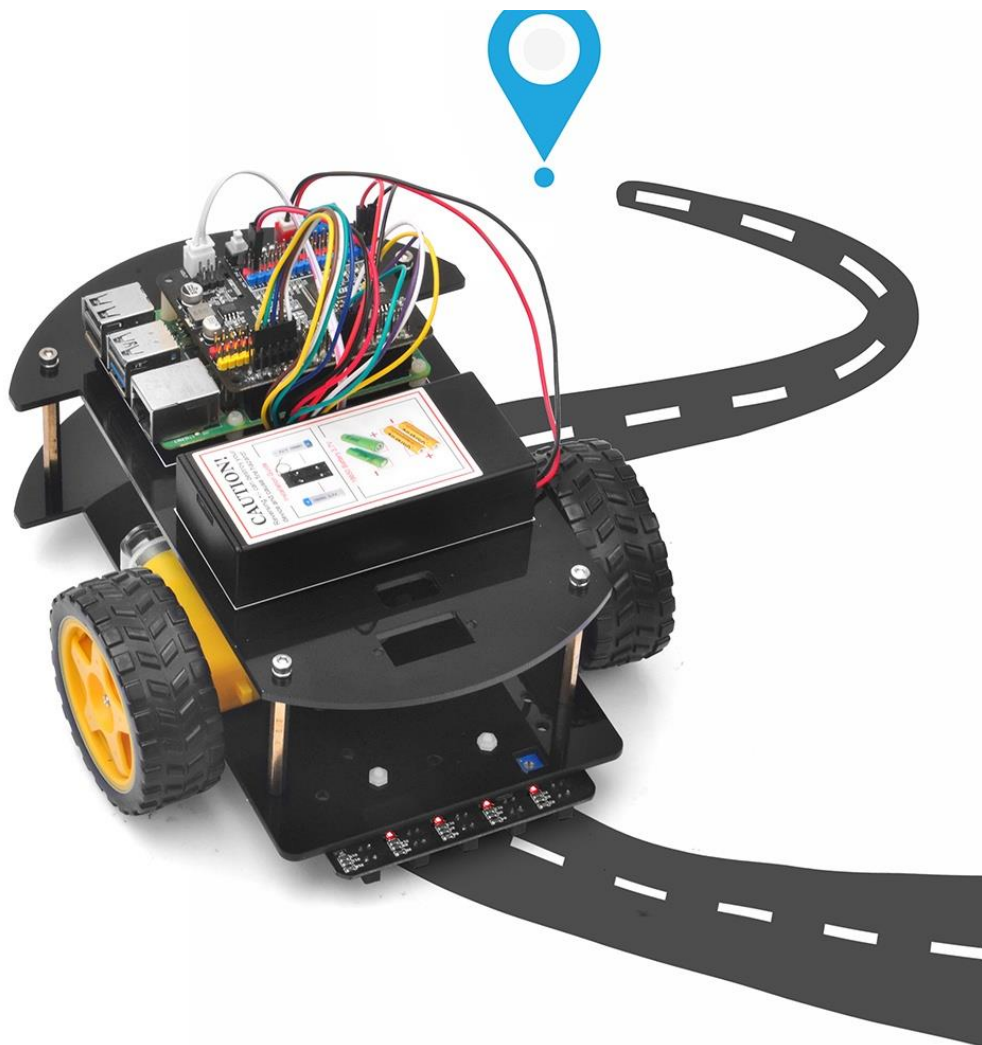
`python picar-line-tracking.py`

If you are using Python 3 in Raspberry Pi 4/3A+, type:

```
python3 picar-line-tracking.py
```

```
pi@raspberrypi:~$ python3 picar-line-tracking.py
Moving servo on channel 0, press Ctrl-C to quit...
00101
00101
00101
00101
00101
00101
00101
00101
```

Your car will move along the black track line.



# OSOY00 Robot Car for Raspberry Pi

## Lesson 3: Obstacle Avoidance



### OBJECTIVE

In this lesson, our Raspberry Pi robot car will use Ultrasonic sensor to detect obstacles and make automatic driving and avoid collision.

Video link: <https://youtu.be/mNMz8uhy8x8>

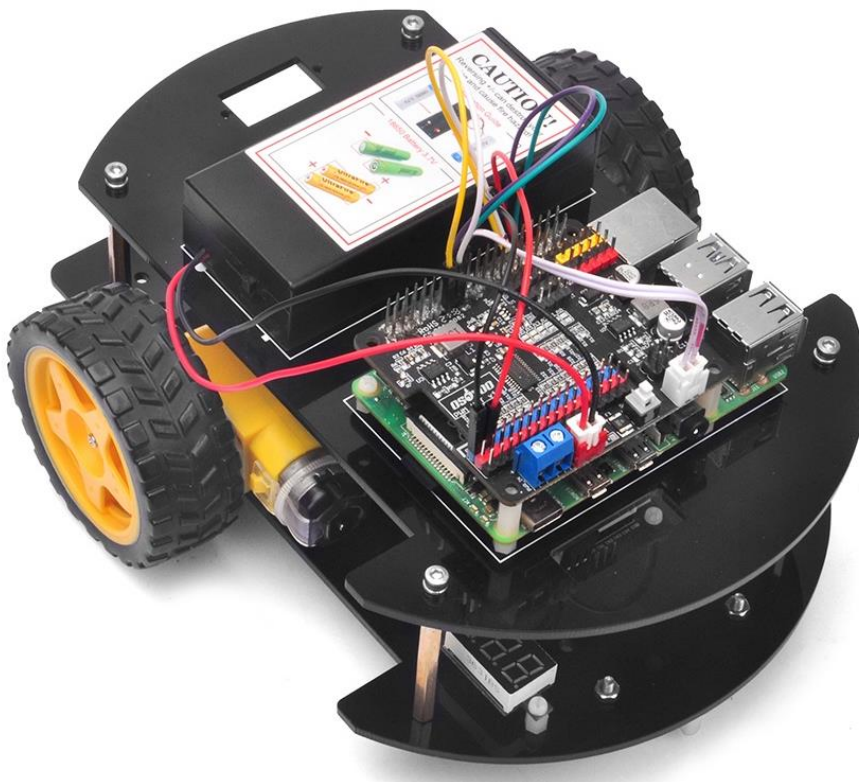
### PARTS & DEVICES

No.	Picture	Device	Qty.	Accessories	Link
1		Ultrasonic Sensor	1	M1.5*8 Screw x 4 M1.5 nut x 4	<a href="#">Click here to buy</a>
2		Servo Motor	1	M2.2*8 Self Tapping Screw x 2 M2*4 Self Tapping Screw x 1	<a href="#">Click here to buy</a>
3		Mount Holder for Ultrasonic Sensor	1	M1.5*8 Screw x 4 M1.5 nut x 4 M2*4 Self Tapping Screw x 1	<a href="#">Click here to buy</a>

4		20Pin jumper wire female to female 20 cm	some		
5		Philips screwdriver	1		<a href="#">Click here to buy</a>

## HARDWARE INSTALLATION

**Step 1:** You must complete [Lesson 1 basic framework](#). If you have completed hardware installation of lesson2, please keep it.



**Step 2:** Install servo motor at the front of upper car chassis with 2pcs M2.2\*8 Self Tapping Screws

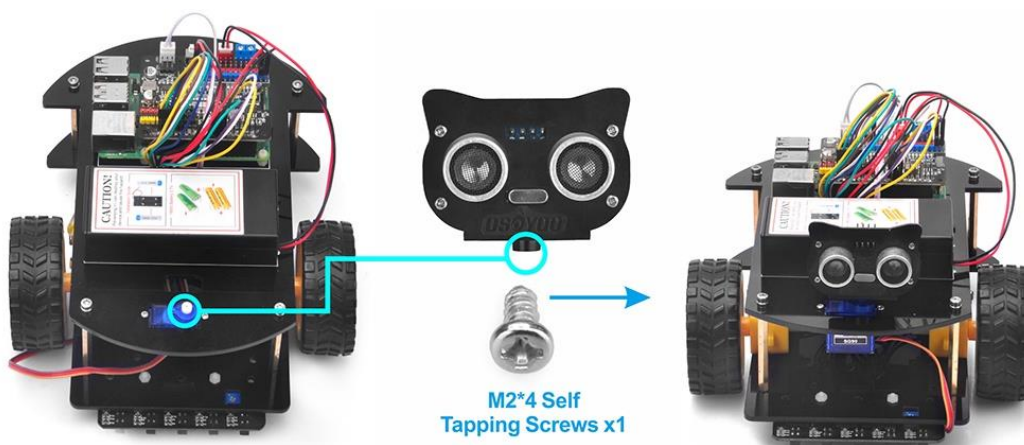




**Step 3:** Install Ultrasonic Module to mount holder with 4pcs M1.5\*8 screw and M1.5 nuts

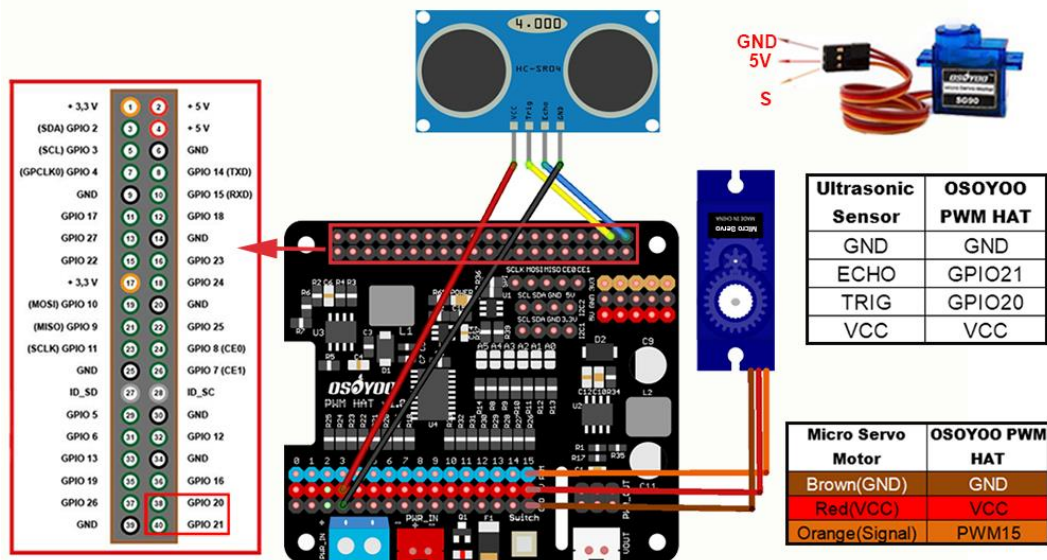


**Step 4:** Install mount holder for Ultrasonic Module on servo motor with M2\*4 Self Tapping screw



CIRCUIT CONNECTION

Connect SG90 servo motor to PWM 15 of PCA9685 compatible module, and connect GND and VCC of ultrasonic module to GND and V+ of PCA9685 compatible module, then connect TRIG and ECHO of ultrasonic module to GPIO20 and GPIO21 of Raspberry Pi as following graph (Remember : DO NOT remove any existing wires installed in Lesson 1) :



## SOFTWARE INSTALLATION

Step 1: Make sure you have installed `rpi.gpio`, `adafruit-pca9685` library and enable I2C in [lesson 1](#).

Step 2: Turn on the robot car and download the python code by typing the following command in your Raspberry Pi terminal:

Wget <http://osoyoo.com/driver/picar/picar-obstacle-avoid.py>

```
pi@raspberrypi:~$ wget http://osoyoo.com/driver/picar/picar-obstacle-avoid.py
--2023-04-19 07:46:39-- http://osoyoo.com/driver/picar/picar-obstacle-avoid.py
Resolving osoyoo.com (osoyoo.com)... 2606:4700:3031::6815:2a5e, 2606:4700:3036::a
c43:a078, 172.67.160.120, ...
Connecting to osoyoo.com (osoyoo.com)|2606:4700:3031::6815:2a5e|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5772 (5.6K) [text/x-python]
Saving to: 'picar-obstacle-avoid.py'

picar-obstacle-avoid 100%[=====>] 5.64K --.-KB/s in 0.003s
2023-04-19 07:46:40 (1.83 MB/s) - 'picar-obstacle-avoid.py' saved [5772/5772]
```

Step 3: Put the robot car on the ground and run the program (**ATTENTION: please select a correct command according to your Pi board**):



If you are using Python 2 in Raspberry Pi 3, type:

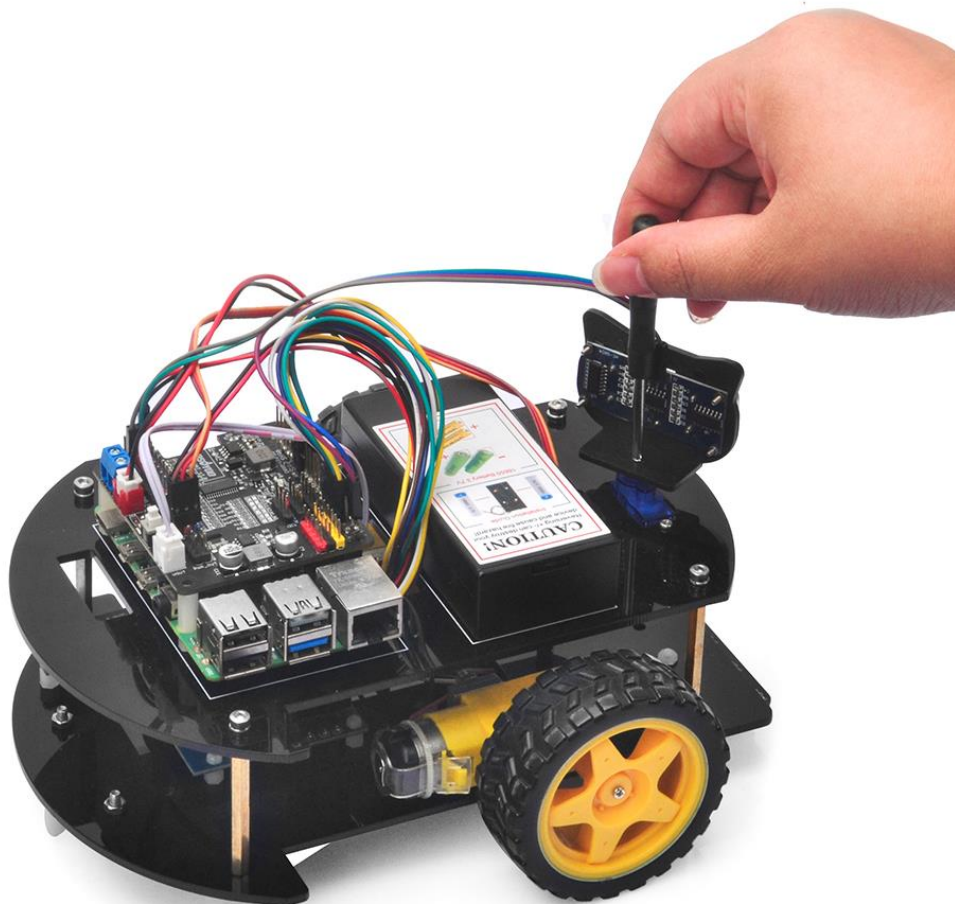
```
python picar-obstacle-avoid.py
```

or if you are using Python 3 in Pi 4/3A+, type :

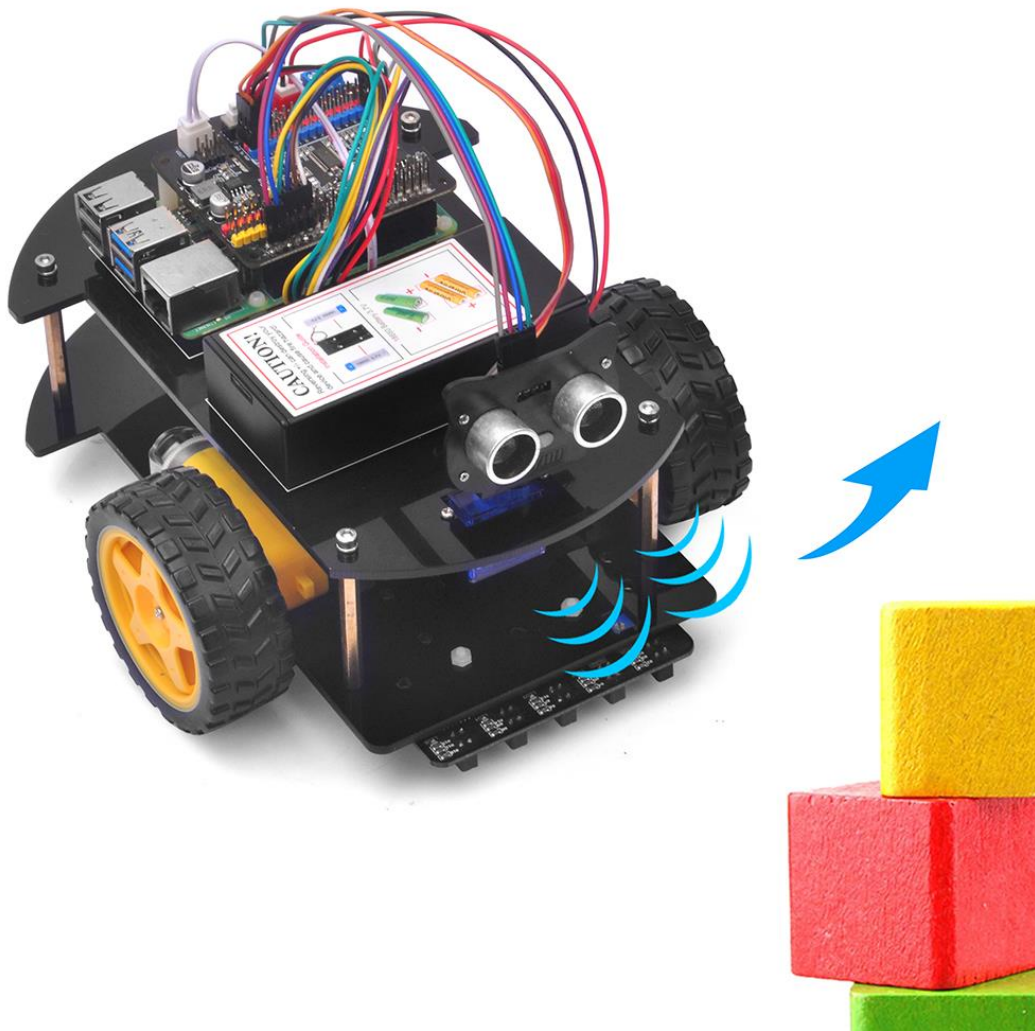
```
python3 picar-obstacle-avoid.py
```

```
pi@raspberrypi:~ $ python3 picar-obstacle-avoid.py  
Moving servo on channel 0, press Ctrl-C to quit...
```

Step 4: After running the last command, the servo motor will make the ultrasonic sensor face left, then right, then middle and stay for a few seconds. If the ultrasonic sensor doesn't face front, please turn off the robot car Immediately. Then remove ultrasonic sensor from servo and re-install it, make sure it faces front and fix it with M2\*4 screw.



Step 5: Turn on the car again, type the same command **`python picar-obstacle-avoid.py`** and run the program again. Now the car will automatically make obstacle avoidance auto driving.



# OSOY00 Robot car for Raspberry Pi

## Lesson 4: Use Mobile APP to control with UDP (Python)



### OBJECTIVE





In this lesson, we will teach you how to use a mobile APP to control a Robot car through UDP protocol. The Raspberry Pi will run a Python program to get a UDP packet from APP.

You will learn how to use Python to get remote data (UDP packet) from mobile APP.

Video Link: <https://youtu.be/7KffHzCofgE>

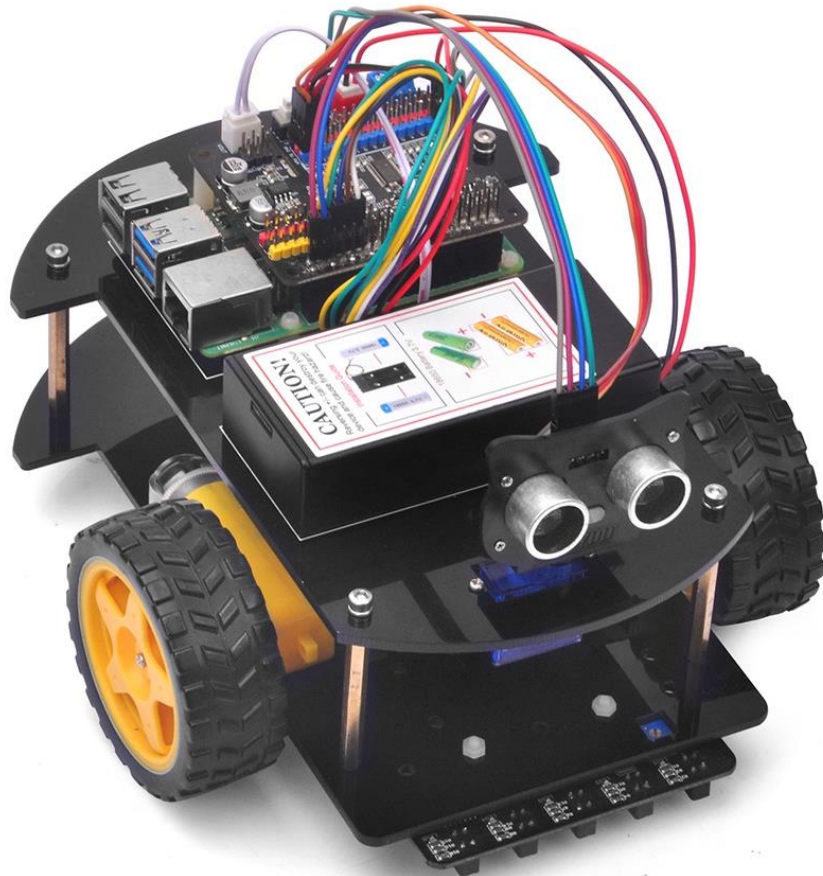
### PARTS & DEVICES

No.	Picture	Device	Qty.	Accessories	Link
1		Ultrasonic Sensor	1	M1.5*8 Screw x 4 M1.5 nut x 4	<a href="#">Click here to buy</a>
2		Servo Motor	1	M2.2*8 Self Tapping Screw x 2 M2*4 Self Tapping Screw x 1	<a href="#">Click here to buy</a>
3		Mount Holder for Ultrasonic Sensor	1	M1.5*8 Screw x 4 M1.5 nut x 4 M2*4 Self Tapping Screw x 1	<a href="#">Click here to buy</a>

4		Tracking sensor module	1	M2.5 Plastic Screw x 2 M2.5 Plastic Nut x2 M2.5 Plastic Pillar x 2	<a href="#">Click here to buy</a>
5		7pin 25 cm Female to Female Cable	1		<a href="#">Click here to buy</a>
6		20Pin jumper wire female to female 20 cm	some		
7		Philips screwdriver	1		<a href="#">Click here to buy</a>

## HARDWARE INSTALLATION

**Step 1:** Install the smart car basic framework as per [lesson 3](#). If you don't install the tracking sensor module, please complete installation in [Lesson 2](#).



## SOFTWARE INSTALLATION

**Step 1)** Make sure you have installed `rpi.gpio` and `adafruit-pca9685` library, and enable I2C in [lesson 1](#).

**Step 2)** Turn on the robot car and type the following command to download the sample code:

```
wget http://osoyoo.com/driver/picar/picar-udp-control.py
```

```
pi@raspberrypi:~ $ wget http://osoyoo.com/driver/picar/picar-udp-control.py
--2023-04-19 07:49:50-- http://osoyoo.com/driver/picar/picar-udp-control.py
Resolving osoyoo.com (osoyoo.com)... 2606:4700:3031::6815:2a5e, 2606:4700:3036::a
c43:a078, 104.21.42.94, ...
Connecting to osoyoo.com (osoyoo.com)[2606:4700:3031::6815:2a5e]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9897 (9.7K) [text/x-python]
Saving to: 'picar-udp-control.py'

picar-udp-control.py 100%[=====] 9.67K --.-KB/s in 0s
2023-04-19 07:49:50 (27.9 MB/s) - 'picar-udp-control.py' saved [9897/9897]
```

**Step 3)** Type the following command to run the sample code (**ATTENTION: please select a correct command according to your Pi board**):

If you are using Python 2 in Raspberry Pi 3, type:

```
python picar-udp-control.py
```

or if you are using Python 3 in Pi 4/3A+, type:

```
python3 picar-udp-control.py
```

```
pi@raspberrypi:~ $ python3 picar-udp-control.py
0
0
0
0
0
0
0
0
0
0
0
```

After above python is running, your car is waiting for command from your cell phone.

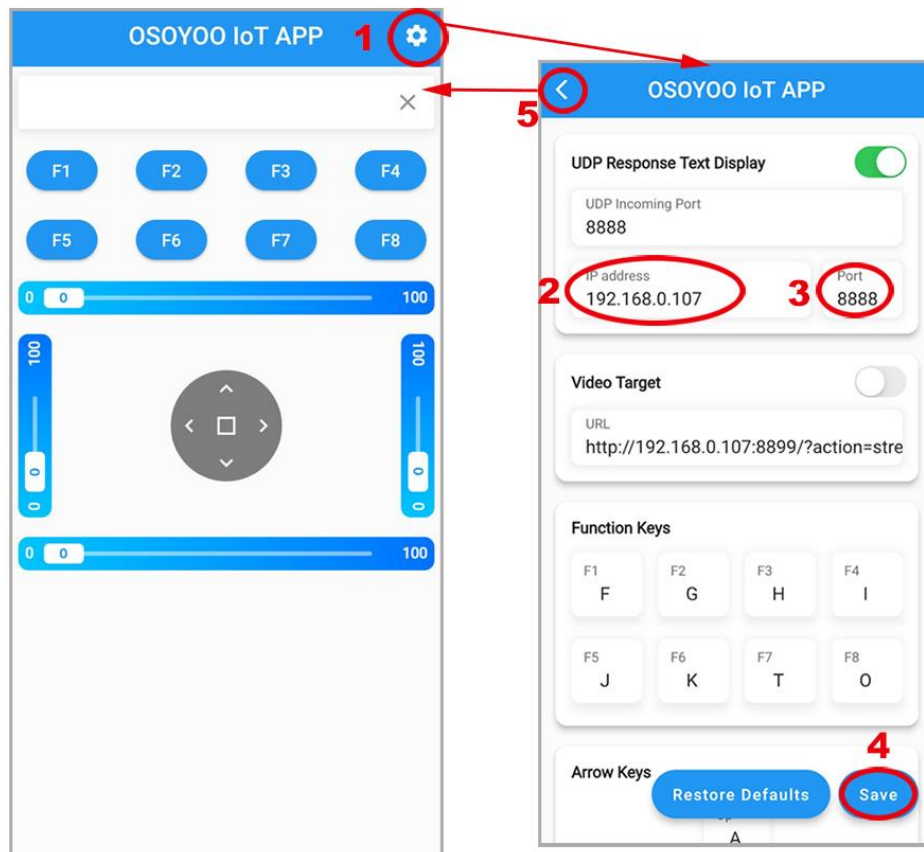
**Step 4)** Download OSOYOO Wi-Fi UDP Robot Car control APP

In Google Play or Apple Store, please search keywords “OSOYOO IoT UDP Robot APP”, you will find a red icon APP as following:





**Step 5)** Connect your phone with the same router Wi-Fi SSID of Raspberry Pi use. Open the APP, click “**settings icon(1)**” to enter setting UI, enter your **Raspberry Pi IP address** and **Port** to **8888**, click **Save**, then click **back icon(5)** to back control UI as following:



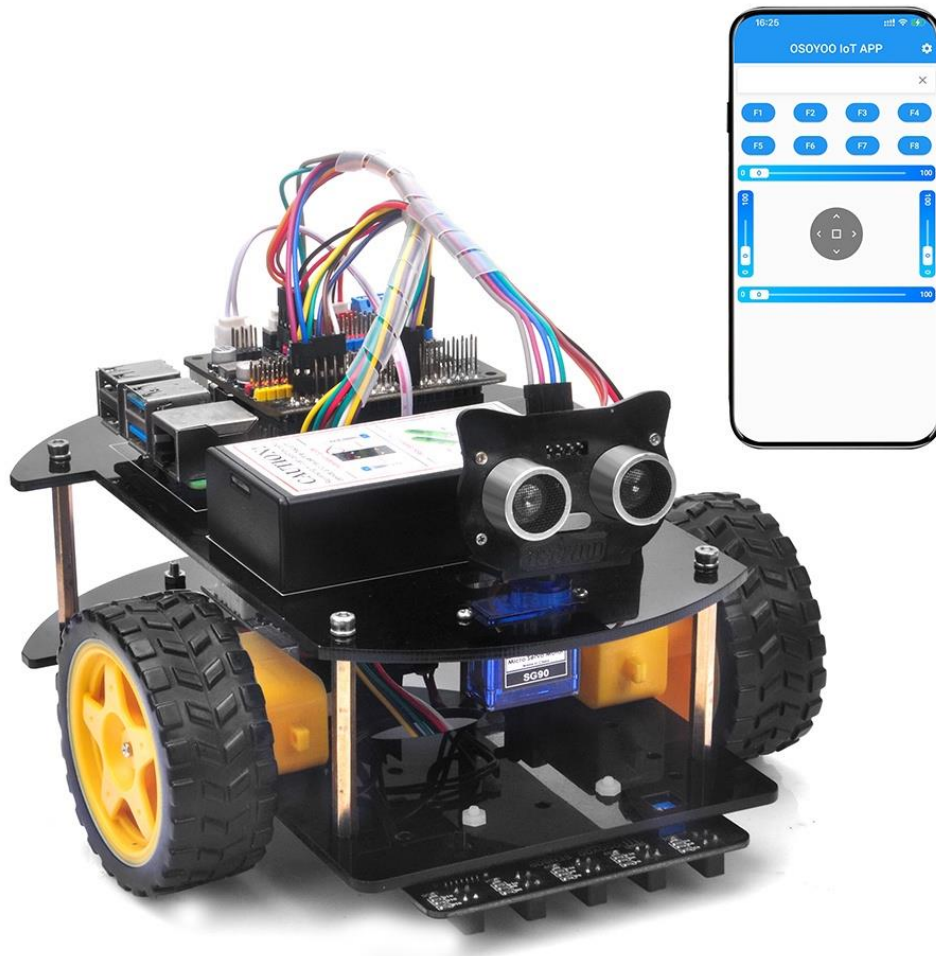
### Step 6)

Now, put the car on the ground, and you can click the < > ^ v direction keys to make the car move. Use || pause key to stop the car movement.

If you click Obstacle key(F8), the car will do obstacle avoidance auto driving similar to [Lesson 3](#)

If you click Tracking key (F7), the car will do link tracking auto driving similar to [lesson 2](#)

Note: F1~F6 are further development functions in the future.



## FAQ about the IoT UDP APP and sketch Code:

**Q1)**How to tune the robot car speed?

**A:** If you want to change the speed performance of the robot car, please change values following parameters in line 19-21 in picar-udp-control.py file :

```
high_speed = 4000 # Max pulse length out of 4096
```

```
mid_speed = 2000 # Max pulse length out of 4096
```

```
low_speed = 1000 # Max pulse length out of 4096
```

**Q 2)**What happened when you press buttons in OSOY00 IoT UDP Robot APP ?

**A:** When you press a button of the APP, the APP will send a single-letter message through UDP protocol to target device Raspberry Pi

Button	UDP message
F1	/
F2	/
F3	/
F4	/
F5	/
F6	/
▲	Advance
▼	Back
▶	Right
◀	Left
square	Stop
F7	Tracking
F8	obstacle

**Q3)**How does Raspberry Pi python program handle the UDP command?

Line 290 to 318 while loop receives UDP data from APP and give it to viable **cur\_status**, ticker function in line 266 – 280 handle the **cur\_status** :

```
def ticker():
    if cur_status=='R':
        turnRight(high_speed, 0)
```

```
if cur_status=='L':  
    turnLeft(0,high_speed)  
if cur_status=='A':  
    forward(mid_speed,mid_speed)  
if cur_status=='B':  
    backward(mid_speed,mid_speed)  
if cur_status=='E':  
    stopcar()  
if cur_status=='T':  
    line_tracking()  
if cur_status=='O':  
    obstacle_avoid()
```

For example, when APP ▲ key is pressed, **cur\_status** value is A, then ticker() function call forward(mid\_speed,mid\_speed) function to make car moving forward.

# OSOYOO Robot car for Raspberry Pi

## Lesson 5: Make a simple website server in Pi (Python)




### OBJECTIVE

There are many ways to control Raspberry Pi Robot Car over Internet.

In this project, we introduce a popular method, which is to install a Web Server on Raspberry Pi. In other words, we can turn our Pi board into a website and use browse or Cell phone APP to exchange data with Raspberry Pi.

Video link: <https://youtu.be/QSMpWBryxAQ>

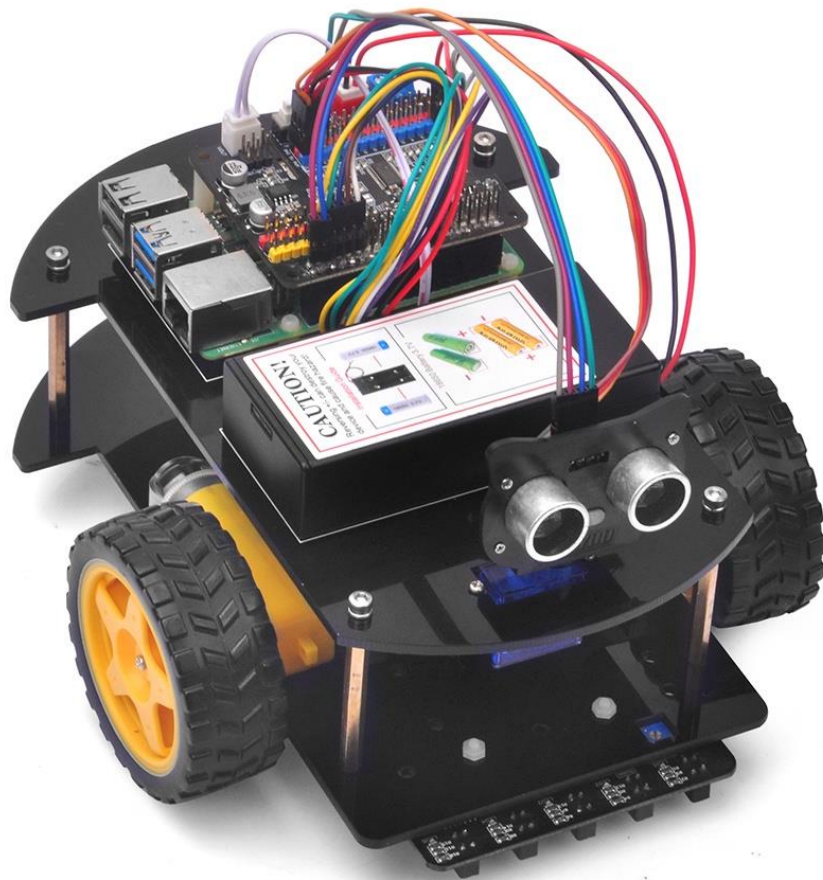
### PARTS & DEVICES

No.	Picture	Device	Qty.	Accessories	Link
1		Tracking sensor module	1	M2.5 Plastic Screw x 2 M2.5 Plastic Nut x2 M2.5 Plastic Pillar x 2	<a href="#">Click here to buy</a>
2		7pin 25 cm Female to Female Cable	1		<a href="#">Click here to buy</a>
3		Philips screwdriver	1		<a href="#">Click here to buy</a>



## HARDWARE INSTALLATION

Step 1: Install the smart car basic framework as per [Lesson 2](#). If you have completed lesson 3, you can just keep it.



## SOFTWARE INSTALLATION

Step 1) Make sure you have installed `rpi.gpio` and `adafruit-pca9685` library, and enable I2C in [lesson 1](#), and power on the robot car.

**Step 2)** Download installation shell script file by typing following command:

```
wget http://osoyoo.com/driver/picar/pi-webserver.sh
```

```
pi@raspberrypi:~ $ wget http://osoyoo.com/driver/picar/pi-webserver.sh
--2023-04-19 07:11:41-- http://osoyoo.com/driver/picar/pi-webserver.sh
Resolving osoyoo.com (osoyoo.com)... 2606:4700:3036::ac43:a078, 2606:4700:3031::6815:2a5e, 172.67.160.120, ...
Connecting to osoyoo.com (osoyoo.com)|2606:4700:3036::ac43:a078|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 138 [text/x-sh]
Saving to: 'pi-webserver.sh'

pi-webserver.sh      100%[=====>]          138  --.-KB/s    in 0s
2023-04-19 07:11:41 (3.53 MB/s) - 'pi-webserver.sh' saved [138/138]
```

**Step 3)** type the following command:

```
bash pi-webserver.sh
```

```
pi@raspberrypi:~ $ bash pi-webserver.sh
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-flask is already the newest version (1.0.2-3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
--2023-04-19 07:12:28-- http://osoyoo.com/driver/picar/pi-webserver.tar.gz
Resolving osoyoo.com (osoyoo.com)... 2606:4700:3036::ac43:a078, 2606:4700:3031::6815:2a5e, 104.21.42.94, ...
Connecting to osoyoo.com (osoyoo.com)|2606:4700:3036::ac43:a078|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 358620 (350K) [application/x-gzip]
Saving to: 'pi-webserver.tar.gz'
```

After running above commands, you will install python Flask library and automatically create a directory called **pi-webserver** and two subdirectories under **pi-webserver** which are **static** and **templates**, the structure is as follows:

```
2023-04-19 07:12:30 (219 KB/s) - 'pi-webserver.tar.gz' saved [358620/358620]

pi-webserver/
pi-webserver/static/
pi-webserver/static/style.css
pi-webserver/static/images/
pi-webserver/static/images/robot52.png
pi-webserver/static/images/lighton.png
pi-webserver/static/images/forward.png
pi-webserver/static/images/lightoff.png
pi-webserver/static/images/left.png
pi-webserver/static/images/right.png
pi-webserver/static/images/reverse.png
pi-webserver/static/images/stop.png
pi-webserver/pi-webserver.py
pi-webserver/templates/
pi-webserver/templates/index.html
```

The **pi-webserver** directory will store our python code file, static directory will store CSS file and templates folder will store HTML files. If you don't know what is HTML, CSS file, it doesn't matter.

If you don't know the IP address of your Pi, please type the command "**hostname -I**" to get your Raspberry Pi's IP, after running the command, you will see following result:

```
pi@raspberrypi:~ $ hostname -I
192.168.0.107 240e:384:847:d300:e65f:1ff:fe3c:7d5e 240e:384:847:d300:f51b:aaec:c5
a9:cd99
```

In the above result, **192.168.0.107** is the IP address of my Raspberry Pi.

**Step 4)** We need to edit python code file and change the IP address in the file. Please type the following commands to enter the pi-webserver folder:

```
cd pi-webserver
```

```
pi@raspberrypi:~ $ cd pi-webserver
pi@raspberrypi:~/pi-webserver $
```

**Step 5)** Please type the following command to edit the file pi-webserver.py

```
nano pi-webserver.py
```

**Step 6)** Please replace "**192.168.0.107**" with your Raspberry Pi IP address, and then click "**Ctrl**" and "**x**" then click "**Y**" to save the file and then click "**enter**" to exit the file.

```
import RPi.GPIO as GPIO
from flask import Flask, render_template, request
app = Flask(__name__)
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
pi_ip_address='192.168.0.107' # replace this ip value with your Raspberry Pi IP$
#define actuators GPIOs
sensor1= 5 # No.1 sensor from right
sensor2= 6 # No.2 sensor from right
sensor3= 13 # middle sensor
```

You will open **pi-webserver.py** in nano editor and see the python code as follows:

```
import RPi.GPIO as GPIO
from flask import Flask, render_template, request
app = Flask(__name__)
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
pi_ip_address='10.0.0.89' # replace this ip value with your Raspberry
Pi IP address, you can use ifconfig command to see the IP

#define actuators GPIOs
sensor1= 25 # No.1 sensor from right
sensor2= 9 # No.2 sensor from right
```

```
sensor3= 11 # middle sensor
sensor4= 8 # No.2 sensor from left
sensor5= 7 #No.1 sensor from left
#initialize GPIO status variables
Sts1 = 0
Sts2 = 0
Sts3 = 0
Sts4 = 0
Sts5 = 0

# Define sensor pins as output
GPIO.setup(sensor1, GPIO.IN)
GPIO.setup(sensor2, GPIO.IN)
GPIO.setup(sensor3, GPIO.IN)
GPIO.setup(sensor4, GPIO.IN)
GPIO.setup(sensor5, GPIO.IN)

@app.route("/")
def index():
    # Read Sensors Status
    Sts1 = GPIO.input(sensor1)
    Sts2 = GPIO.input(sensor2)
    Sts3 = GPIO.input(sensor3)
    Sts4 = GPIO.input(sensor4)
    Sts5 = GPIO.input(sensor5)

    templateData = {
        'title' : 'IR tracking sensor Status!',
        'sensor_status_1' : Sts1,
        'sensor_status_2' : Sts2,
        'sensor_status_3' : Sts3,
        'sensor_status_4' : Sts4,
        'sensor_status_5' : Sts5
    }
    return render_template('index.html', **templateData)

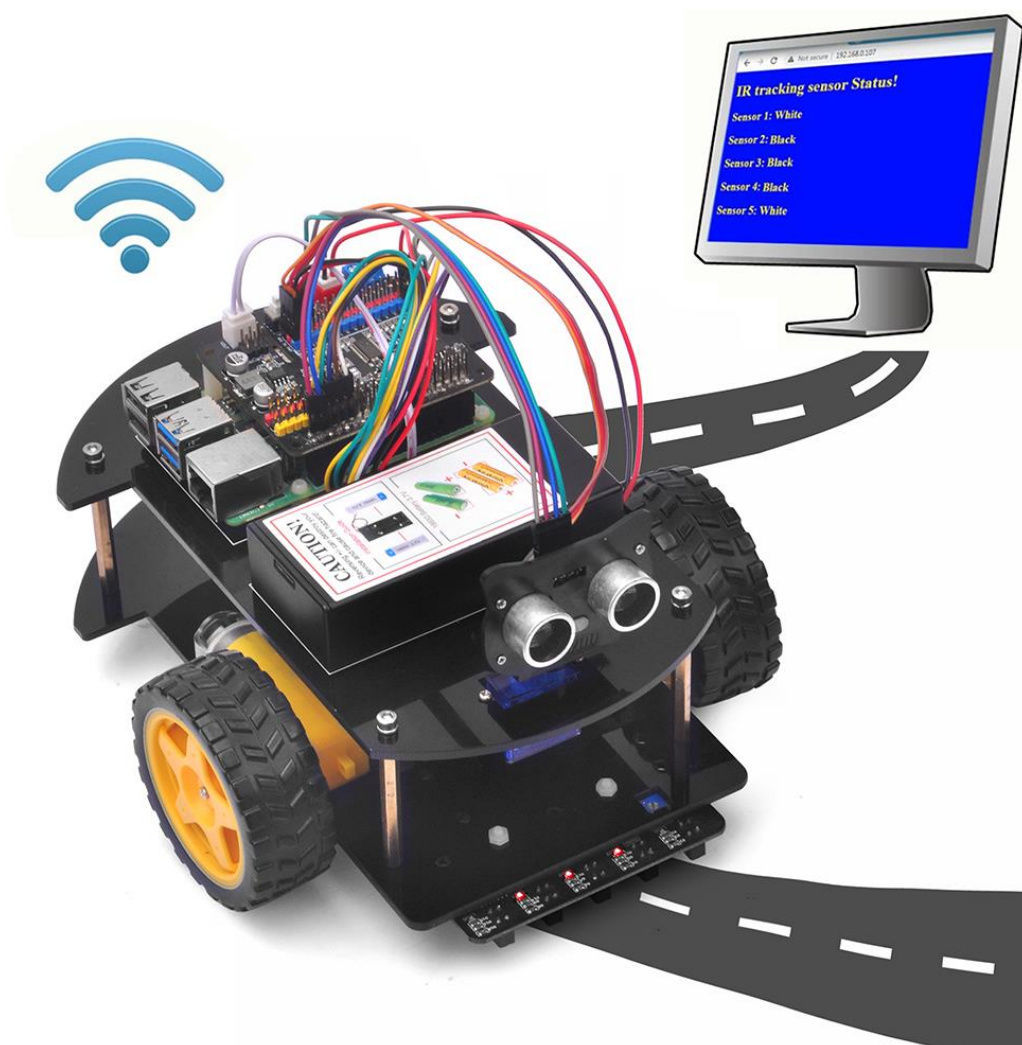
if __name__ == "__main__":
    app.run(host=pi_ip_address, port=80, debug=True)
```

**Step 7)** Run the code by typing the following command:

```
sudo pythonpi-webserver.py
```

```
pi@raspberrypi:~/pi-webserver $ sudo python pi-webserver.py
* Serving Flask app "pi-webserver" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://192.168.0.107:80/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 544-420-453
```

**Step 8)** Your python flask web server will start. Now in your PC or cell phone which is the same Wi-Fi network of your Raspberry Pi, open browser and visit [http://your\\_raspberry\\_pi\\_ip\\_address](http://your_raspberry_pi_ip_address) (in my case <http://192.168.0.107>), you will see your 5 tracking sensors status in your browser:



The sensor status will be updated every 5 seconds, so if you put the black line under other sensors, the sensor value in the above web page will change after 5 seconds.



# OSOY00 Robot Car for Raspberry Pi




## Lesson 6: Web-Camera-Controlled IoT Raspberry Pi Robot Car (Python)


### OBJECTIVE

In this lesson, we will show you how to use Python3 Flask and M-Jpeg Streamer software to control a Raspberry Pi Robot Car through the Internet. You will monitor the car's real-time movement through its eye (front camera).

Video Link: <https://youtu.be/AmVGTx2xVtI>

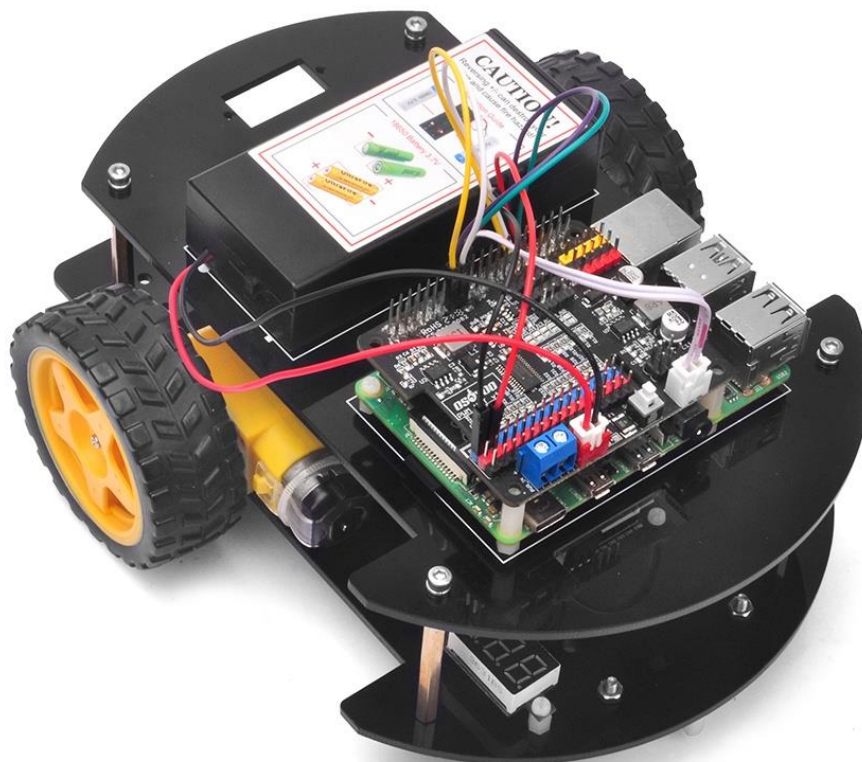
### PARTS & DEVICES

No.	Picture	Device	Qty.	Accessories	Link
1		CSI camera	1	M2 push pin rivets x 4	<a href="#">Click here to buy</a>
2		Servo Motor	1	M2.2*8 Self Tapping Screw x 2 M2*4 Self Tapping Screw x 1	<a href="#">Click here to buy</a>
3		Mount Holder for CSI camera	1	M2 push pin rivets x 4 M1.5*6 Self Tapping Screws x2 M2*4 Self Tapping Screw x 1	<a href="#">click here to buy</a>

4		Philips screwdriver	1		<a href="#">Click here to buy</a>
---	---	---------------------	---	--	-----------------------------------

## HARDWARE INSTALLATION

**Step 1:** You must complete [Lesson 1 basic framework](#). If You have completed [lesson 3](#) before, please remove the Ultrasonic sensor mount holder from servo motor and wires from Ultrasonic sensor, then continue on with this lesson



**Step 2:** Install servo motor at the front of upper car chassis with 2pcs M2.2\*8 Self Tapping Screws



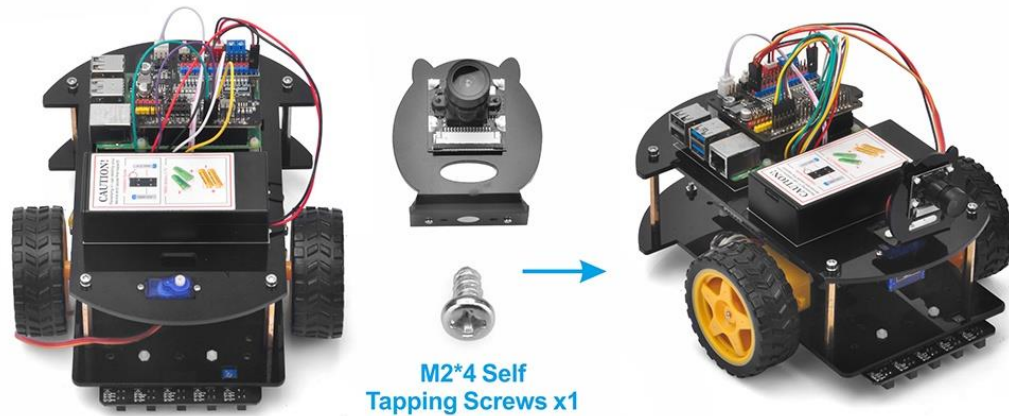
**Step 3:** Choose slotted bracket in servo motor to cross Camera holder from top to bottom and fix blade with M1.5\*6 Self Tapping Screws



**Step 4:** Install CSI camera to holder with 4pcs M2 push pin rivets

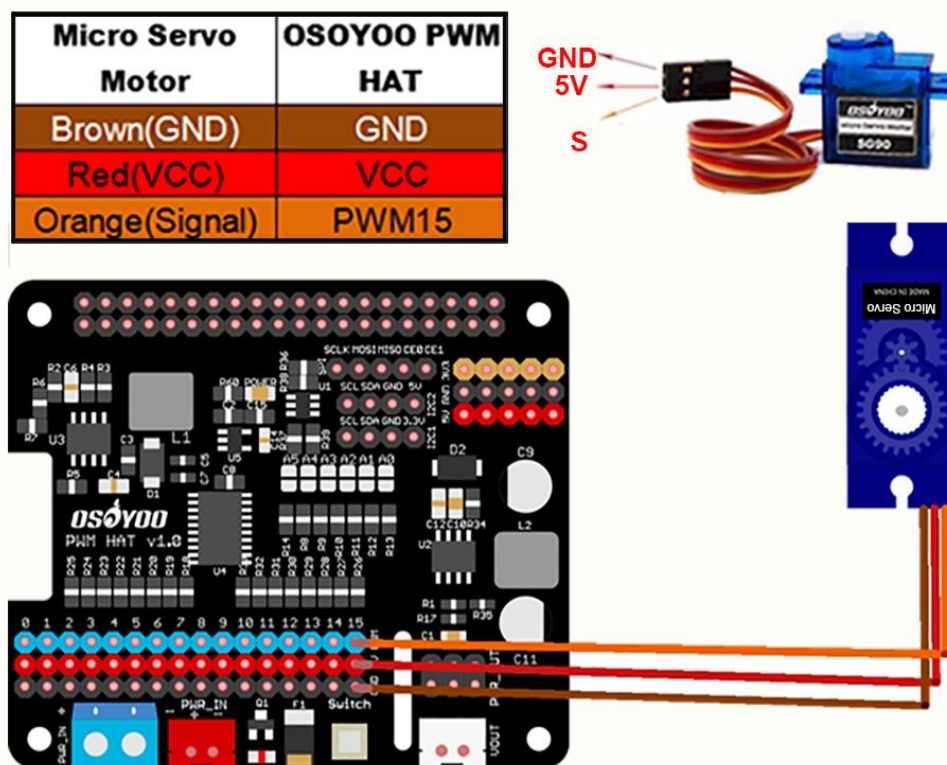


**Step 5:** Install Camera holder on servo motor with 1pc M2\*4 Self Tapping Screws



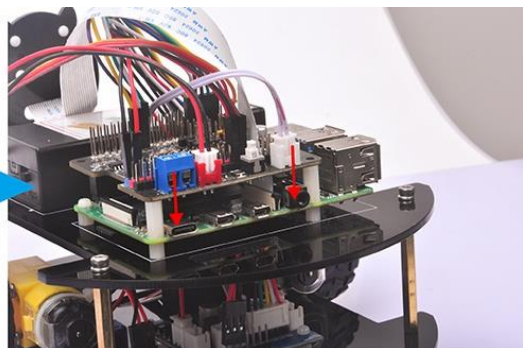
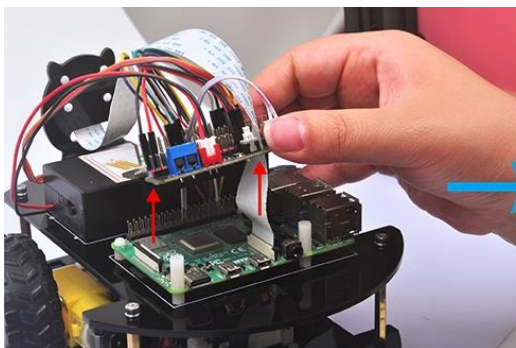
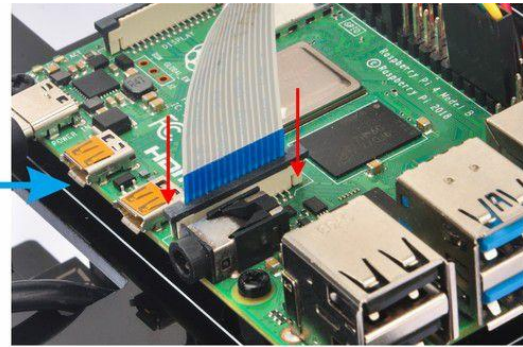
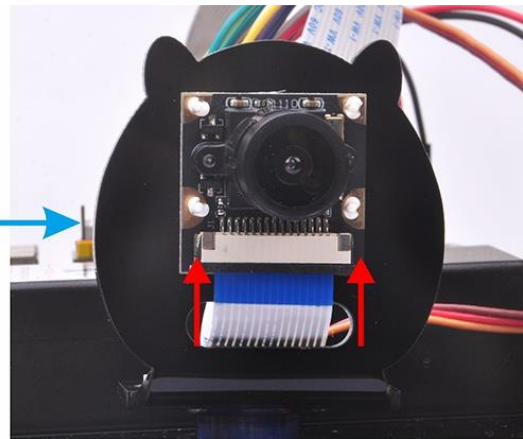
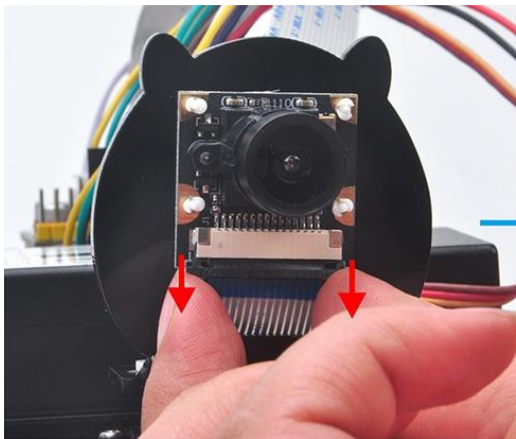
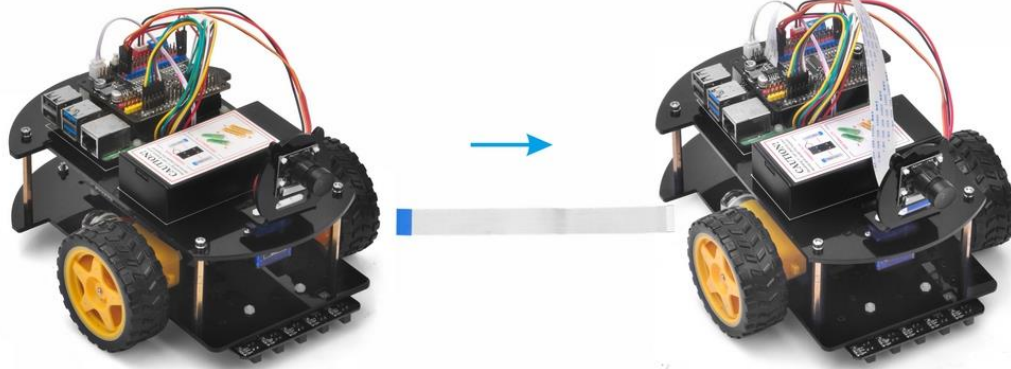
## CIRCUIT CONNECTION

**Step 1)** Servo Must be installed and connected PCA9685 PWM 15 port



**Step 2)** Connect CSI camera to CSI Slot of Raspberry Pi with CSI ribbon cable (Please pay attention to the connections of the cable before you install it.)





## SOFTWARE INSTALLATION



## Start jpeg-streamer server in your Raspberry Pi

Step 1: Make sure you have installed `rpi.gpio`, `adafruit-pca9685` library and enable I2C in [lesson 1](#), and power on the robot car.

**Step 2:** Please enable Camera in Raspberry Pi by typing following command

```
sudo raspi-config
```

Then select → Interfacing Options→ Camera→ Yes→ Ok→ Finish (Please reboot the Raspberry Pi according to the notice.)

```
Raspberry Pi Software Configuration Tool (raspi-config)

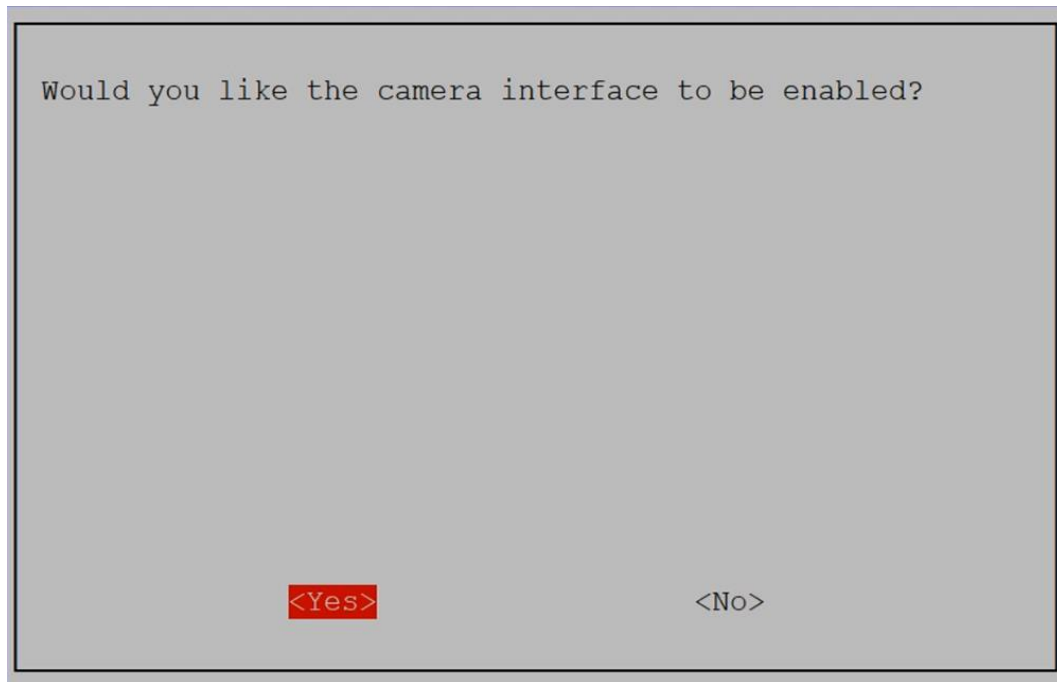
1 System Options      Configure system settings
2 Display Options     Configure display settings
3 Interface Options   Configure connections to peripherals
4 Performance Options Configure performance settings
5 Localisation Options Configure language and regional settings
6 Advanced Options    Configure advanced settings
8 Update              Update this tool to the latest version
9 About raspi-config  Information about this configuration tool

<Select>                <Finish>
```

```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera      Enable/disable connection to the Raspberry Pi Camera
P2 SSH         Enable/disable remote command line access using SSH
P3 VNC         Enable/disable graphical remote access using RealVNC
P4 SPI         Enable/disable automatic loading of SPI kernel module
P5 I2C         Enable/disable automatic loading of I2C kernel module
P6 Serial Port Enable/disable shell messages on the serial connection
P7 1-Wire      Enable/disable one-wire interface
P8 Remote GPIO Enable/disable remote access to GPIO pins

<Select>                <Back>
```



**Step 3:** Type following command to install mjpeg-streamer software :

`wget https://osoyoo.com/driver/picar/camininstall.sh`

```
pi@raspberrypi:~ $ wget https://osoyoo.com/driver/picar/camininstall.sh
--2023-04-19 07:21:47-- https://osoyoo.com/driver/picar/camininstall.sh
Resolving osoyoo.com (osoyoo.com)... 2606:4700:3036::ac43:a078, 2606:4700:3031::6815:2a5e, 172.67.160.120, ...
Connecting to osoyoo.com (osoyoo.com)|2606:4700:3036::ac43:a078|:443... connected
.
HTTP request sent, awaiting response... 200 OK
Length: 219 [text/x-sh]
Saving to: 'camininstall.sh'

camininstall.sh      100%[=====>]          219  --.-KB/s    in 0s
2023-04-19 07:21:49 (53.2 MB/s) - 'camininstall.sh' saved [219/219]
```

**Step 4:** Type the following command:

`wget https://osoyoo.com/driver/picar/startcam.sh`

```
pi@raspberrypi:~ $ wget https://osoyoo.com/driver/picar/startcam.sh
--2023-04-19 07:22:20-- https://osoyoo.com/driver/picar/startcam.sh
Resolving osoyoo.com (osoyoo.com)... 2606:4700:3031::6815:2a5e, 2606:4700:3036::ac43:a078, 172.67.160.120, ...
Connecting to osoyoo.com (osoyoo.com)|2606:4700:3031::6815:2a5e|:443... connected
.
HTTP request sent, awaiting response... 200 OK
Length: 202 [text/x-sh]
Saving to: 'startcam.sh'

startcam.sh          100%[=====>]          202  --.-KB/s    in 0s
2023-04-19 07:22:22 (51.2 MB/s) - 'startcam.sh' saved [202/202]
```

**Step 5:** the following command at last:

`bash caminstall.sh`

```
pi@raspberrypi:~ $ bash caminstall.sh
Hit:1 http://archive.raspberrypi.org/debian buster InRelease
Hit:2 http://raspbian.raspberrypi.org/raspbian buster InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  cmake-data libjpeg9 libjsoncpp1 librhash0 libuv1
Suggested packages:
  cmake-doc ninja-build
The following NEW packages will be installed:
  cmake cmake-data libjpeg9 libjpeg9-dev libjsoncpp1 librhash0 libuv1
0 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.
Need to get 4,742 kB of archives.
After this operation, 23.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

After running above commands, mjpeg-streamer software is installed in your Raspberry Pi.  
So Simple!

**Step 6:** Start jpeg-streamer server in your Raspberry Pi

Run following command in your Pi terminal will start your mjpeg-streamer server

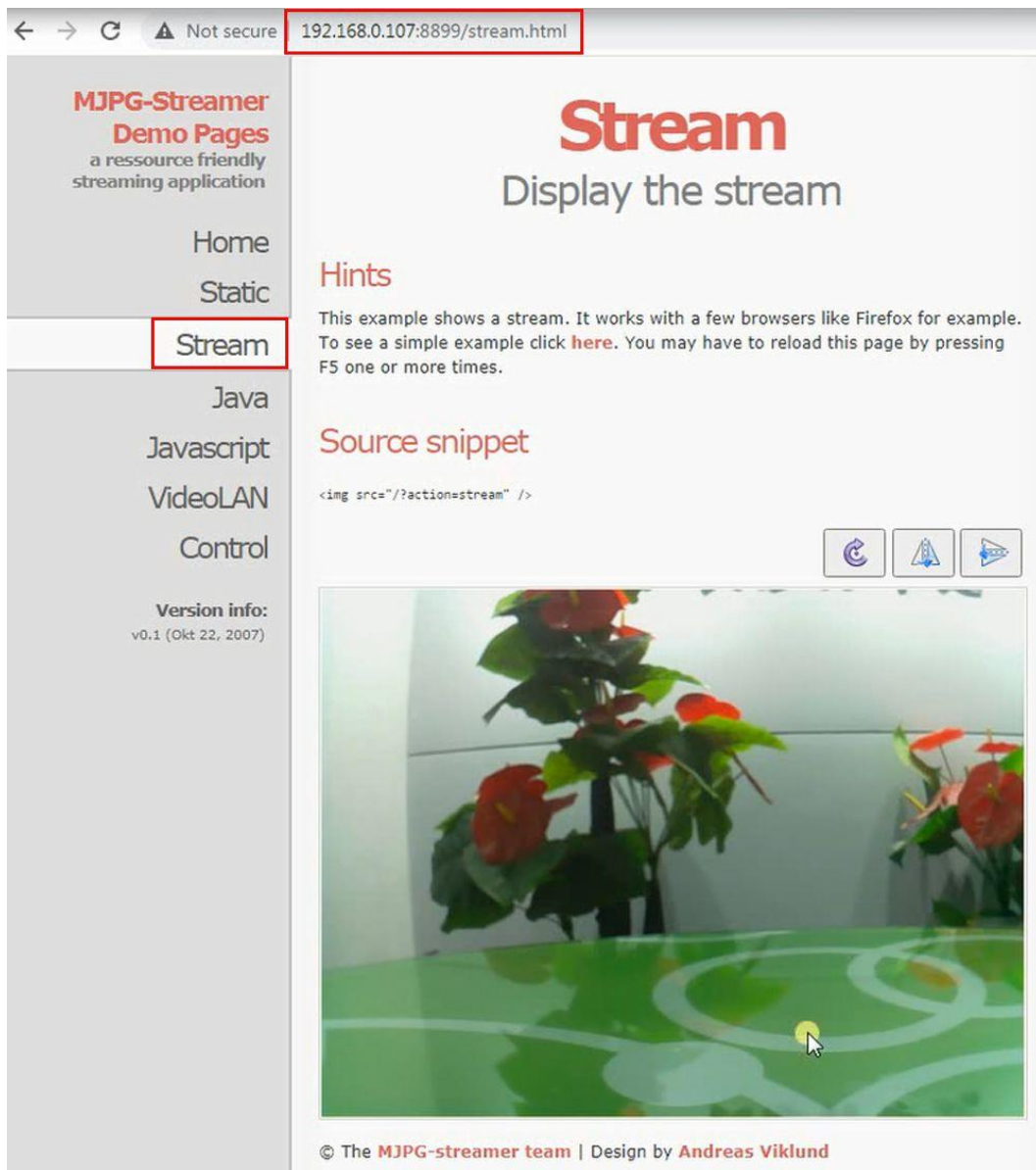
`bash startcam.sh`

```
pi@raspberrypi:~ $ bash camstart.sh
MJPEG Streamer Version: svn rev: Unversioned directory
i: Using V4L2 device.: /dev/video0
i: Desired Resolution: 640 x 480
i: Frames Per Second.: 30
i: Format.....: YUV
i: JPEG Quality.....: 80
o: www-folder-path...: www/
o: HTTP TCP port.....: 8899
o: username:password.: disabled
o: commands.....: enabled
```

These message means your video server is running at port 8899 in your Pi.

**Step 7:** Now we can use your browser to test the surveillance video. Now please visit  
[http://your\\_raspberry\\_pi\\_ip:8899](http://your_raspberry_pi_ip:8899) (in my case <http://192.168.0.32:8899>),

click stream link or direct visit [http://your\\_raspberry\\_pi\\_ip:8899/?action=stream](http://your_raspberry_pi_ip:8899/?action=stream), you will see the video surveillance.



## Web-Camera-Controlled IoT Robot Car

### METHOD 1: USE BROWSER TO CONTROL A RASPBERRY PI ROBOT CAR THROUGH THE INTERNET

**Step 1:** Open a new terminal window and run the following code to install the OSOYOO web camera controlled robot car software



```
wget http://osoyoo.com/driver/picar/osoyoowebcar.sh
```

```
pi@raspberrypi:~ $ wget http://osoyoo.com/driver/picar/osoyoowebcar.sh
--2023-04-19 07:26:28-- http://osoyoo.com/driver/picar/osoyoowebcar.sh
Resolving osoyoo.com (osoyoo.com)... 2606:4700:3031::6815:2a5e, 2606:4700:3036::a
c43:a078, 172.67.160.120, ...
Connecting to osoyoo.com (osoyoo.com)|2606:4700:3031::6815:2a5e|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 138 [text/x-sh]
Saving to: 'osoyoowebcar.sh'

osoyoowebcar.sh      100%[=====>]          138  --.-KB/s   in 0s

2023-04-19 07:26:29 (3.55 MB/s) - 'osoyoowebcar.sh' saved [138/138]
```

**Step 2:** type the following command:

```
bash osoyoowebcar.sh
```

```
pi@raspberrypi:~ $ bash osoyoowebcar.sh
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-flask is already the newest version (1.0.2-3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
--2023-04-19 07:27:11-- http://osoyoo.com/driver/picar/osoyoowebcar.tar.gz
Resolving osoyoo.com (osoyoo.com)... 2606:4700:3036::ac43:a078, 2606:4700:3031::6
815:2a5e, 104.21.42.94, ...
Connecting to osoyoo.com (osoyoo.com)|2606:4700:3036::ac43:a078|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 359899 (351K) [application/x-gzip]
Saving to: 'osoyoowebcar.tar.gz'

osoyoowebcar.tar.gz  28%[====>              ] 100.67K   224KB/s
```

After running above commands, you will automatically create a directory called **osoyoowebcar**, the structure is as follows:

```
2023-04-19 07:27:13 (427 KB/s) - 'osoyoowebcar.tar.gz' saved [359899/359899]

osoyoowebcar/
osoyoowebcar/webcar.py
osoyoowebcar/templates/
osoyoowebcar/templates/index.html
osoyoowebcar/static/
osoyoowebcar/static/style.css
osoyoowebcar/static/images/
osoyoowebcar/static/images/lightoff.png
osoyoowebcar/static/images/lighton.png
osoyoowebcar/static/images/robot52.png
osoyoowebcar/static/images/forward.png
osoyoowebcar/static/images/left.png
osoyoowebcar/static/images/reverse.png
osoyoowebcar/static/images/right.png
osoyoowebcar/static/images/stop.png
```

**Step 3:** type the following command to enter the folder osoyoowebcar

```
cd osoyoowebcar
```

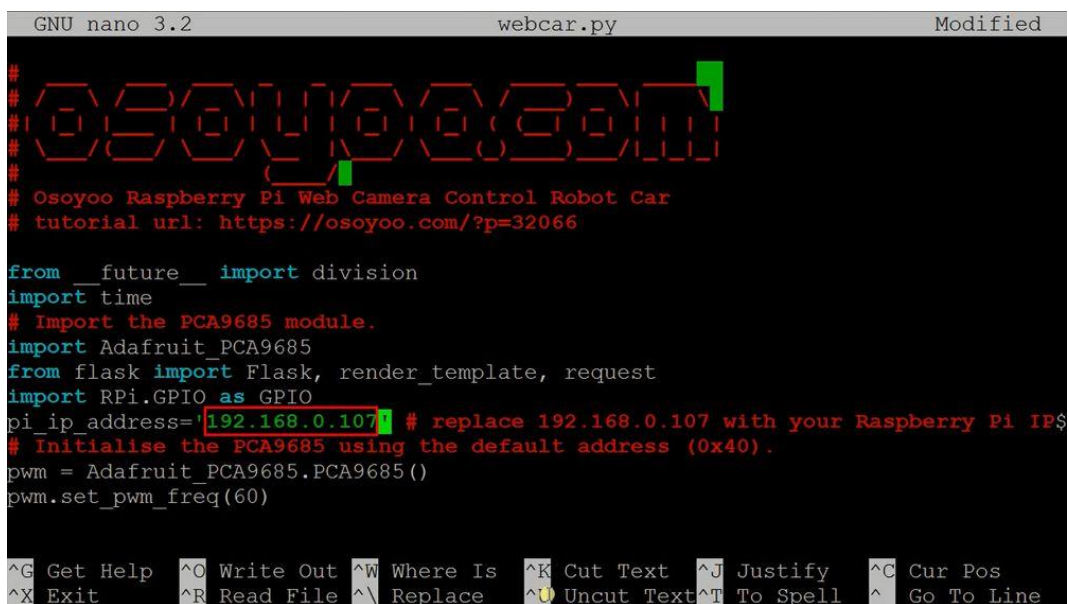


```
pi@raspberrypi:~ $ cd osoyoowebcar
pi@raspberrypi:~/osoyoowebcar $
```

**Step 4:** Then type the following command to edit the file webcar.py

```
nano webcar.py
```

**Step 5:** Replace ip address **192.168.0.107** in **line 15** to your Raspberry Pi IP address, and click “**Ctrl**+”**X**” then “**Y**” to save the file and then click “**enter**” to exit the file.



```
GNU nano 3.2 webcar.py Modified
#
# Osoyoo Raspberry Pi Web Camera Control Robot Car
# tutorial url: https://osoyoo.com/?p=32066
#
from __future__ import division
import time
# Import the PCA9685 module.
import Adafruit_PCA9685
from flask import Flask, render_template, request
import RPi.GPIO as GPIO
pi_ip_address='192.168.0.107' # replace 192.168.0.107 with your Raspberry Pi IP$
# Initialise the PCA9685 using the default address (0x40).
pwm = Adafruit_PCA9685.PCA9685()
pwm.set_pwm_freq(60)

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

**Step 6:** Then type the following command to edit the file templates/index.html

```
nano templates/index.html
```

**Step 7:** Please change **192.168.0.107** in line **34** to your pi’s IP address, and click “**Ctrl**+”**X**” then “**Y**” to save the file and then click “**enter**” to exit the file

```
GNU nano 3.2 templates/index.html Modified
<div style="text-align:center">
<h1> OSOY00 Web Car in Raspberry Pi</h1>
<br><br>
<div class="slidecontainer">
  <p>Camera Direction Slider</p>
  Left<input type="range" min="1" max="179" value="90" id="myRange">Right<br>
  <div id="demo"></div>
</div>
<br><br>
<iframe src="http://192.168.0.107:8899/?action=stream" frameborder="0" align="m$
<br><br>
<br><br>
<span style="display:inline-block;padding:5px;border:1px solid #fc0; font-size:$
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^T Uncut Text ^_ To Spell ^_ Go To Line
```

**Step 8:** put the car on the ground and run the code by typing the following command  
(**ATTENTION:** please select a correct command according to your Pi board):

If you are using Python 2 in Raspberry Pi 3, type:

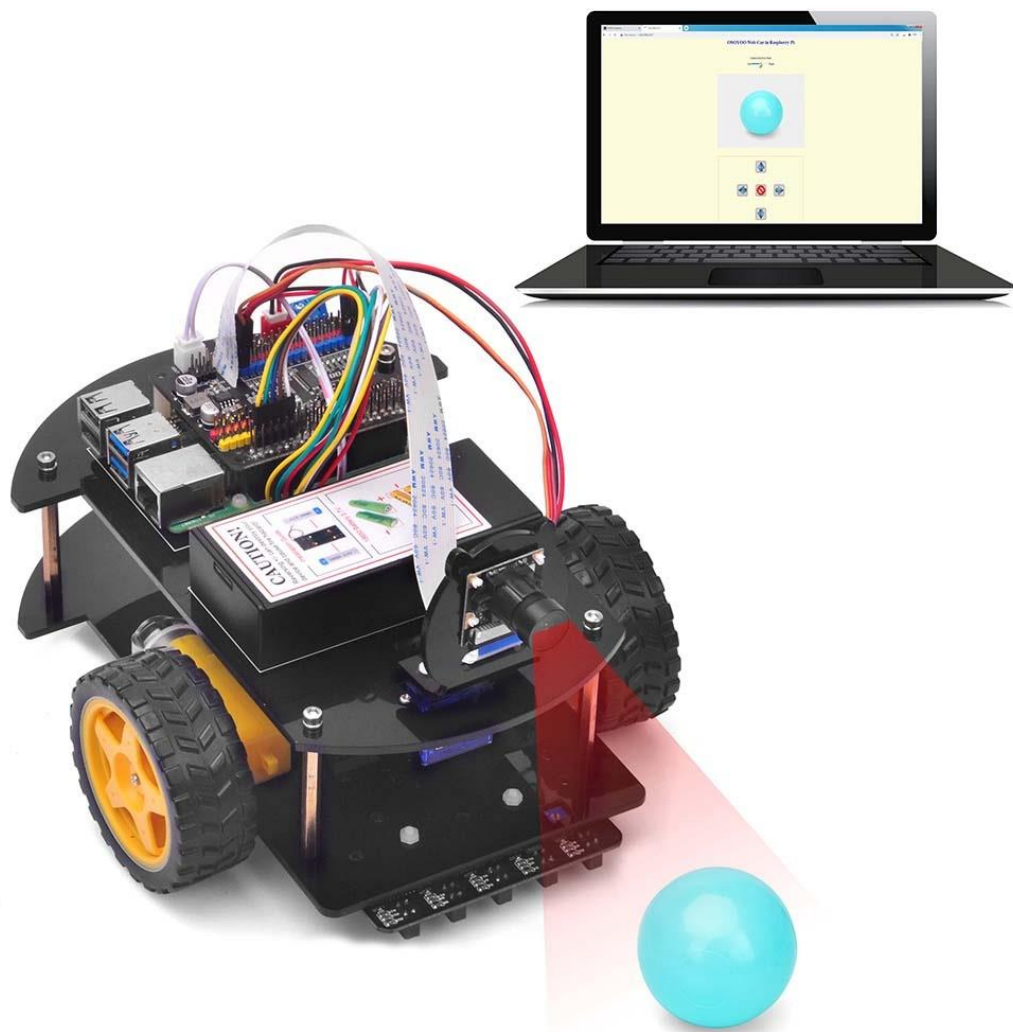
```
sudo python2 ~/osoyoowebcar/webcar.py
```

If you are using Python 3 in Raspberry Pi 4/3A+, type:

```
sudo python3 ~/osoyoowebcar/webcar.py
```

```
pi@raspberrypi:~/osoyoowebcar $ sudo python3 ~/osoyoowebcar/webcar.py
* Serving Flask app "webcar" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://192.168.0.107:80/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 251-590-641
```

**Step 9:** In your PC or cell phone which is the same Wi-Fi network of your Raspberry Pi, open the browser and visit [http://your\\_RaspberryPi\\_ip\\_address](http://your_RaspberryPi_ip_address).



The arrows buttons are direction control keys, the red circle button in the middle is the STOP key. There are four speed control buttons in the bottom:

- 0 key means the slowest speed
- — key means the 2nd slow speed
- == key means regular speed
- ++ key means fastest speed

In the top of the page, there is a Camera Direction Slider, you can move the slider in order to rotate the servo and change the camera orientation.

## METHOD 2: USE APP TO CONTROL A RASPBERRY PI ROBOT CAR THROUGH THE INTERNET

**Step 1:** Open a new terminal window and run the following code to download the sample code for APP control

```
wget http://osoyoo.com/driver/picar/udp-camera-car.py
```

```
pi@raspberrypi:~ $ wget http://osoyoo.com/driver/picar/udp-camera-car.py
--2023-04-25 05:02:56-- http://osoyoo.com/driver/picar/udp-camera-car.py
Resolving osoyoo.com (osoyoo.com)... 2606:4700:3036::ac43:a078, 2606:4700:3031::
6815:2a5e, 172.67.160.120, ...
Connecting to osoyoo.com (osoyoo.com)|2606:4700:3036::ac43:a078|:80... connected
.
HTTP request sent, awaiting response... 200 OK
Length: 8700 (8.5K) [text/x-python]
Saving to: 'udp-camera-car.py'

udp-camera-car.py 100%[=====>] 8.50K --.-KB/s in 0s
2023-04-25 05:02:57 (27.2 MB/s) - 'udp-camera-car.py' saved [8700/8700]
```

**Step 2:** Type the following command to run the sample code (**ATTENTION: please select a correct command according to your Pi board**):

If you are using Python 2 in Raspberry Pi 3, type:

```
python udp-camera-car.py
```

or if you are using Python 3 in Pi 4/3A+, type:

```
python3 udp-camera-car.py
```

```
pi@raspberrypi:~ $ python3 udp-camera-car.py
b''
b''
b'ST=10'
b'ST=29'
b'A'
b'B'
b'L'
b'E'
b'B'
b'E'
b'R'
b'E'
```

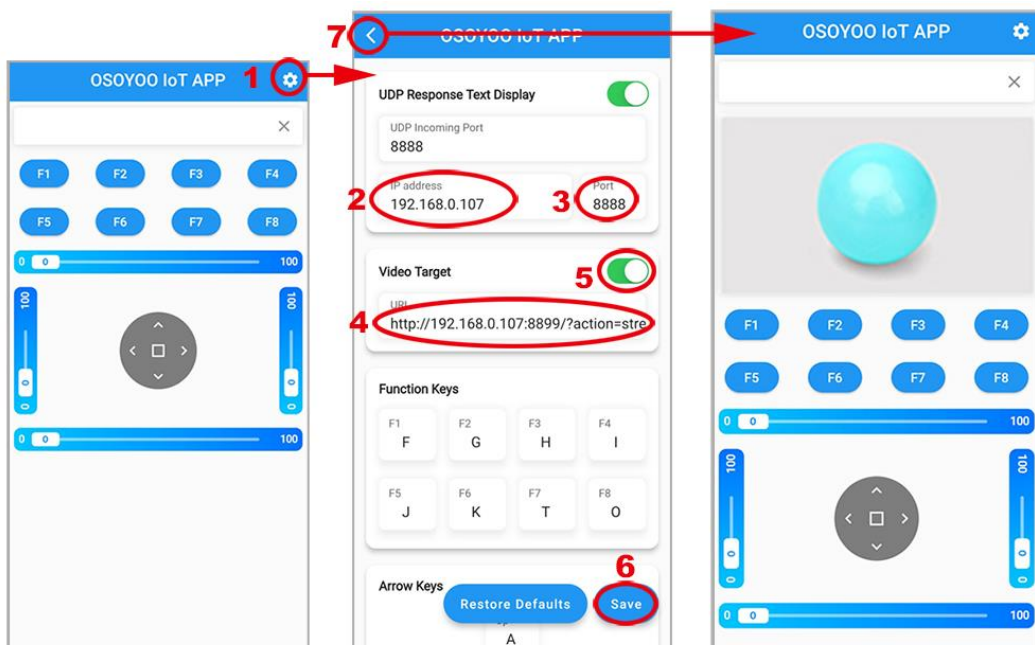
**Step 3)** Download OSOYOO Wi-Fi UDP Robot Car control APP

In Google Play or Apple Store, please search keywords “OSOYOO IoT UDP Robot APP”,

you will find a red icon APP as following:



**Step 4)** Connect your phone with the same router Wi-Fi SSID of Raspberry Pi use. Open the APP, click “**setting icon(1)**” to enter “setting UI”, enter IP address to your **Raspberry Pi IP address** and **Port** to **8888**, enter **http://your\_raspberry\_pi\_ip:8899/?action=stream** to **video Target**, then turn **Video Target switch** to green, then click **save**, then click “**back icon(7)**” to back control UI as following:



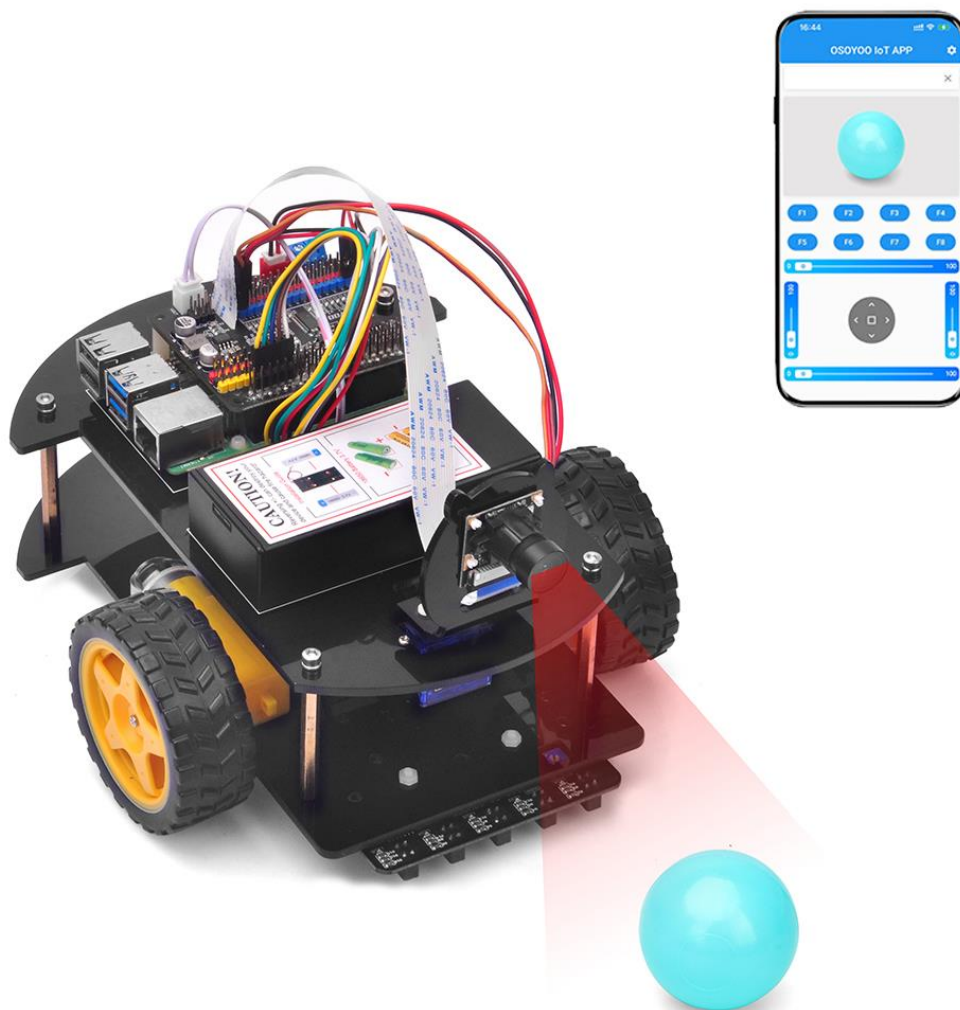


**Step 5)** Now, put the car on the ground, and you can click the < > ^ v direction keys to make the car move. Use || pause key to stop the car movement.

The Slider at the top of direction keys is a Camera Direction Slider, and you can move the slider in order to rotate the servo and change the camera orientation.

The slider at the left of direction keys is a speed slider, and you can move the slider up to speed up this robot.

Note: F1~F8, and the bottom and right slider are unavailable in this lesson.



#### Reference:

**Marcelo Rovai** : Python WebServer With Flask and Raspberry Pi

<https://towardsdatascience.com/python-webserver-with-flask-and-raspberry-pi-398423cc6f5d>